SAMBA Project Documentation

SAMBA Team

samba@samba.org

Edited by

John H Terpstra

Jelmer Vernooij

Gerald (Jerry) Carter

SAMBA Project Documentation

by SAMBA Team

Edited by John H Terpstra

Edited by Jelmer Vernooij

Edited by Gerald (Jerry) Carter

Published Sunday 6 April

This book is a collection of HOWTOs added to Samba documentation over the years. Samba is always under development, and so is it's documentation. The most recent version of this document can be found at http://www.samba.org/ on the "Documentation" page. Please send updates to jerry@samba.org (mailto:jerry@samba.org) or jelmer@samba.org (mailto:jelmer@samba.org).

Table of Contents

I. General installation

Introduction

This part contains general info on how to install samba and how to configure the parts of samba you will most likely need. PLEASE read this.

Chapter 1. How to Install and Test SAMBA

1.1. Obtaining and installing samba

Binary packages of samba are included in almost any Linux or Unix distribution. There are also some packages available at the samba homepage (http://samba.org/).

If you need to compile samba from source, check the appropriate appendix chapter.

1.2. Configuring samba

Samba's configuration is stored in the smb.conf file, that usually resides in `/etc/samba/smb.conf` or `/usr/local/samba/lib/smb.conf`. You can either edit this file yourself or do it using one of the many graphical tools that are available, such as the web-based interface swat, that is included with samba.

1.2.1. Editing the `smb.conf` file

There are sample configuration files in the examples subdirectory in the distribution. I suggest you read them carefully so you can see how the options go together in practice. See the man page for all the options.

The simplest useful configuration file would be something like this:

```
[global]
workgroup = MYGROUP

[homes]
guest ok = no
read only = no
```

which would allow connections by anyone with an account on the server, using either their login name or "**homes**" as the service name. (Note that I also set the workgroup that Samba is part of. See BROWSING.txt for details)

Make sure you put the `smb.conf` file in the same place you specified in the`Makefile` (the default is to look for it in `/usr/local/samba/lib/`).

For more information about security settings for the **[homes]** share please refer to the chapter Securing Samba.

1.2.1.1. Test your config file with testparm

It's important that you test the validity of your `smb.conf` file using the testparm program. If testparm runs OK then it will list the loaded services. If not it will give an error message.

Make sure it runs OK and that the services look reasonable before proceeding.

Always run testparm again when you change `smb.conf`!

1.2.2. SWAT

SWAT is a web-based interface that helps you configure samba. SWAT might not be available in the samba package on your platform, but in a seperate package. Please read the swat manpage on compiling, installing and configuring swat from source.

To launch SWAT just run your favorite web browser and point it at "http://localhost:901/". Replace *`localhost`* with the name of the computer you are running samba on if you are running samba on a different computer then your browser.

Note that you can attach to SWAT from any IP connected machine but connecting from a remote machine leaves your connection open to password sniffing as passwords will be sent in the clear over the wire.

1.3. Try listing the shares available on your server

```
$ smbclient -L yourhostname
```

You should get back a list of shares available on your server. If you don't then something is incorrectly setup. Note that this method can also be used to see what shares are available on other LanManager clients (such as WfWg).

If you choose user level security then you may find that Samba requests a password before it will list the shares. See the **smbclient** man page for details. (you can force it to list the shares without a password by adding the option -U% to the command line. This will not work with non-Samba servers)

1.4. Try connecting with the unix client

```
$ smbclient  //yourhostname/aservice
```

Typically the `yourhostname` would be the name of the host where you installed smbd. The `aservice` is any service you have defined in the `smb.conf` file. Try your user name if you just have a **[homes]** section in `smb.conf`.

For example if your unix host is `bambi` and your login name is `fred` you would type:

```
$ smbclient //bambi/fred
```

1.5. Try connecting from a DOS, WfWg, Win9x, WinNT, Win2k, OS/2, etc... client

Try mounting disks. eg:

```
C:\WINDOWS\> net use d: \\servername\service
```

Try printing. eg:

```
C:\WINDOWS\> net use lpt1: \\servername\spoolservice
```

```
C:\WINDOWS\> print filename
```

1.6. What If Things Don't Work?

Then you might read the file chapter Diagnosis and the FAQ. If you are still stuck then try the mailing list or newsgroup (look in the README for details). Samba has been successfully installed at thousands of sites worldwide, so maybe someone else has hit your problem and has overcome it. You could also use the WWW site to scan back issues of the samba-digest.

When you fix the problem *please* send some updates of the documentation (or source code) to one of the documentation maintainers or the list.

1.6.1. Scope IDs

By default Samba uses a blank scope ID. This means all your windows boxes must also have a blank scope ID. If you really want to use a non-blank scope ID then you will need to use the 'netbios scope' smb.conf option. All your PCs will need to have the same setting for this to work. I do not recommend scope IDs.

1.6.2. Locking

One area which sometimes causes trouble is locking.

There are two types of locking which need to be performed by a SMB server. The first is "record locking" which allows a client to lock a range of bytes in a open file. The second is the "deny modes" that are specified when a file is open.

Record locking semantics under Unix is very different from record locking under Windows. Versions of Samba before 2.2 have tried to use the native fcntl() unix system call to implement proper record locking between different Samba clients. This can not be fully correct due to several reasons. The simplest is the fact that a Windows client is allowed to lock a byte range up to 2^32 or 2^64, depending on the client OS. The unix locking only supports byte ranges up to 2^31. So it is not possible to correctly satisfy a lock request above 2^31. There are many more differences, too many to be listed here.

Samba 2.2 and above implements record locking completely independent of the underlying unix system. If a byte range lock that the client requests happens to fall into the range 0-2^31, Samba hands this request down to the Unix system. All other locks can not be seen by unix anyway.

Strictly a SMB server should check for locks before every read and write call on a file. Unfortunately with the way fcntl() works this can be slow and may overstress the rpc.lockd. It is also almost always unnecessary as clients are supposed to independently make locking calls before reads and writes anyway if locking is important to them. By default Samba only makes locking calls when explicitly asked to by a client, but if you set "strict locking = yes" then it will make lock checking calls on every read and write.

You can also disable by range locking completely using "locking = no". This is useful for those shares that don’t support locking or don’t need it (such as cdroms). In this case Samba fakes the return codes of locking calls to tell clients that everything is OK.

The second class of locking is the "deny modes". These are set by an application when it opens a file to determine what types of access should be allowed simultaneously with its open. A client may ask for DENY_NONE, DENY_READ, DENY_WRITE or DENY_ALL. There are also special compatibility modes called DENY_FCB and DENY_DOS.

Chapter 2. Quick Cross Subnet Browsing / Cross Workgroup Browsing guide

This document should be read in conjunction with Browsing and may be taken as the fast track guide to implementing browsing across subnets and / or across workgroups (or domains). WINS is the best tool for resolution of NetBIOS names to IP addeses. WINS is NOT involved in browse list handling except by way of name to address mapping.

> **Note:** MS Windows 2000 and later can be configured to operate with NO NetBIOS over TCP/IP. Samba-3 and later also supports this mode of operation.

2.1. Discussion

Firstly, all MS Windows networking is based on SMB (Server Message Block) based messaging. SMB messaging may be implemented using NetBIOS or without NetBIOS. Samba implements NetBIOS by encapsulating it over TCP/IP. MS Windows products can do likewise. NetBIOS based networking uses broadcast messaging to affect browse list management. When running NetBIOS over TCP/IP this uses UDP based messaging. UDP messages can be broadcast or unicast.

Normally, only unicast UDP messaging can be forwarded by routers. The **remote announce** parameter to smb.conf helps to project browse announcements to remote network segments via unicast UDP. Similarly, the **remote browse sync** parameter of `smb.conf` implements browse list collation using unicast UDP.

Secondly, in those networks where Samba is the only SMB server technology wherever possible nmbd should be configured on one (1) machine as the WINS server. This makes it easy to manage the browsing environment. If each network segment is configured with it's own Samba WINS server, then the only way to get cross segment browsing to work is by using the **remote announce** and the **remote browse sync** parameters to your `smb.conf` file.

If only one WINS server is used for an entire multi-segment network then the use of the **remote announce** and the **remote browse sync** parameters should NOT be necessary.

As of Samba 3 WINS replication is being worked on. The bulk of the code has been committed, but it still needs maturation.

Right now samba WINS does not support MS-WINS replication. This means that when setting up Samba as a WINS server there must only be one nmbd configured as a WINS server on the network. Some sites have used multiple Samba WINS servers for redundancy (one server per subnet) and then used **remote browse sync** and **remote announce** to affect browse list collation across all segments. Note that this means clients will only resolve local names, and must be configured to use DNS to resolve names on other subnets in order to resolve the IP addresses of the servers they can see on other subnets. This setup is not recommended, but is mentioned as a practical consideration (ie: an 'if all else fails' scenario).

Lastly, take note that browse lists are a collection of unreliable broadcast messages that are repeated at intervals of not more than 15 minutes. This means that it will take time to establish a browse list and it can take up to 45 minutes to stabilise, particularly across network segments.

2.2. How browsing functions and how to deploy stable and dependable browsing using Samba

As stated above, MS Windows machines register their NetBIOS names (i.e.: the machine name for each service type in operation) on start up. Also, as stated above, the exact method by which this name registration takes place is determined by whether or not the MS Windows client/server has been given a WINS server address, whether or not LMHOSTS lookup is enabled, or if DNS for NetBIOS name resolution is enabled, etc.

In the case where there is no WINS server all name registrations as well as name lookups are done by UDP broadcast. This isolates name resolution to the local subnet, unless LMHOSTS is used to list all names and IP addresses. In such situations Samba provides a means by which the samba server name may be forcibly injected into the browse list of a remote MS Windows network (using the **remote announce** parameter).

Where a WINS server is used, the MS Windows client will use UDP unicast to register with the WINS server. Such packets can be routed and thus WINS allows name resolution to function across routed networks.

During the startup process an election will take place to create a local master browser if one does not already exist. On each NetBIOS network one machine will be elected to function as the domain master browser. This domain browsing has nothing to do with MS security domain control. Instead, the domain master browser serves the role of contacting each local master browser (found by asking WINS or from LMHOSTS) and exchanging browse list contents. This way every master browser will eventually obtain a complete list of all machines that are on the network. Every 11-15 minutes an election is held to determine which machine will be the master browser. By the nature of the election criteria used, the machine with the highest uptime, or the most senior protocol version, or other criteria, will win the election as domain master browser.

Clients wishing to browse the network make use of this list, but also depend on the availability of correct name resolution to the respective IP address/addresses.

Any configuration that breaks name resolution and/or browsing intrinsics will annoy users because they will have to put up with protracted inability to use the network services.

Samba supports a feature that allows forced synchonisation of browse lists across routed networks using the **remote browse sync** parameter in the `smb.conf` file. This causes Samba to contact the local master browser on a remote network and to request browse list synchronisation. This effectively bridges two networks that are separated by routers. The two remote networks may use either broadcast based name resolution or WINS based name resolution, but it should be noted that the **remote browse sync** parameter provides browse list synchronisation - and that is distinct from name to address resolution, in other words, for cross subnet browsing to function correctly it is essential that a name to address resolution mechanism be provided. This mechanism could be via DNS, `/etc/hosts`, and so on.

2.3. Use of the Remote Announce parameter

The **remote announce** parameter of `smb.conf` can be used to forcibly ensure that all the NetBIOS names on a network get announced to a remote network. The syntax of the **remote announce** parameter is:

```
remote announce = a.b.c.d [e.f.g.h] ...
```

_or_

```
remote announce = a.b.c.d/WORKGROUP [e.f.g.h/WORKGROUP] ...
```

where:

`a.b.c.d` and *`e.f.g.h`*

is either the LMB (Local Master Browser) IP address or the broadcst address of the remote network. ie: the LMB is at 192.168.1.10, or the address could be given as 192.168.1.255 where the netmask is assumed to be 24 bits (255.255.255.0). When the remote announcement is made to the broadcast address of the remote network every host will receive our announcements. This is noisy and therefore undesirable but may be necessary if we do NOT know the IP address of the remote LMB.

`WORKGROUP`

is optional and can be either our own workgroup or that of the remote network. If you use the workgroup name of the remote network then our NetBIOS machine names will end up looking like they belong to that workgroup, this may cause name resolution problems and should be avoided.

2.4. Use of the Remote Browse Sync parameter

The **remote browse sync** parameter of `smb.conf` is used to announce to another LMB that it must synchronise it's NetBIOS name list with our Samba LMB. It works ONLY if the Samba server that has this option is simultaneously the LMB on it's network segment.

The syntax of the **remote browse sync** parameter is:

```
remote browse sync = a.b.c.d
```

where *`a.b.c.d`* is either the IP address of the remote LMB or else is the network broadcast address of the remote segment.

2.5. Use of WINS

Use of WINS (either Samba WINS _or_ MS Windows NT Server WINS) is highly recommended. Every NetBIOS machine registers it's name together with a name_type value for each of of several types of service it has available. eg: It registers it's name directly as a unique (the type 0x03) name. It also registers it's name if it is running the lanmanager compatible server service (used to make shares and printers available to other users) by registering the server (the type 0x20) name.

All NetBIOS names are up to 15 characters in length. The name_type variable is added to the end of the name - thus creating a 16 character name. Any name that is shorter than 15 characters is padded with spaces to the 15th character. ie: All NetBIOS names are 16 characters long (including the name_type information).

WINS can store these 16 character names as they get registered. A client that wants to log onto the network can ask the WINS server for a list of all names that have registered the NetLogon service name_type. This saves broadcast traffic and greatly expedites logon processing. Since broadcast name resolution can not be used across network segments this type of information can only be provided via WINS _or_ via statically configured `lmhosts` files that must reside on all clients in the absence of WINS.

WINS also serves the purpose of forcing browse list synchronisation by all LMB's. LMB's must synchronise their browse list with the DMB (domain master browser) and WINS helps the LMB to identify it's DMB. By definition this will work only within a single workgroup. Note that the domain master browser has NOTHING to do with what is referred to as an MS Windows NT Domain. The later is a reference to a security environment while the DMB refers to the master controller for browse list information only.

Use of WINS will work correctly only if EVERY client TCP/IP protocol stack has been configured to use the WINS server/s. Any client that has not been configured to use the WINS server will continue to use only broadcast based name registration so that WINS may NEVER get to know about it. In any case, machines that have not registered with a WINS server will fail name to address lookup attempts by other clients and will therefore cause workstation access errors.

To configure Samba as a WINS server just add **wins support = yes** to the `smb.conf` file [globals] section.

To configure Samba to register with a WINS server just add "wins server = a.b.c.d" to your smb.conf file [globals] section.

> **Important:** Never use both **wins support = yes** together with **wins server = a.b.c.d** particularly not using it's own IP address. Specifying both will cause nmbd to refuse to start!

2.6. Do NOT use more than one (1) protocol on MS Windows machines

A very common cause of browsing problems results from installing more than one protocol on an MS Windows machine.

Every NetBIOS machine takes part in a process of electing the LMB (and DMB) every 15 minutes. A set of election criteria is used to determine the order of precidence for winning this election process. A machine running Samba or Windows NT will be biased so that the most suitable machine will predictably win and thus retain it's role.

The election process is "fought out" so to speak over every NetBIOS network interface. In the case of a Windows 9x machine that has both TCP/IP and IPX installed and has NetBIOS enabled over both protocols the election will be decided over both protocols. As often happens, if the Windows 9x machine is the only one with both protocols then the LMB may be won on the NetBIOS interface over the IPX protocol. Samba will then lose the LMB role as Windows 9x will insist it knows who the LMB is. Samba will then cease to function as an LMB and thus browse list operation on all TCP/IP only machines will fail.

Windows 95, 98, 98se, Me are referred to generically as Windows 9x. The Windows NT4, 2000, XP and 2003 use common protocols. These are roughly referred to as the WinNT family, but it should be recognised that 2000 and XP/2003 introduce new protocol extensions that cause them to behave differently from MS Windows NT4. Generally, where a server does NOT support the newer or extended protocol, these will fall back to the NT4 protocols.

The safest rule of all to follow it this - USE ONLY ONE PROTOCOL!

2.7. Name Resolution Order

Resolution of NetBIOS names to IP addresses can take place using a number of methods. The only ones that can provide NetBIOS name_type information are:

WINS: the best tool!
LMHOSTS: is static and hard to maintain.
Broadcast: uses UDP and can not resolve names across remote segments.

Alternative means of name resolution includes:

/etc/hosts: is static, hard to maintain, and lacks name_type info
DNS: is a good choice but lacks essential name_type info.

Many sites want to restrict DNS lookups and want to avoid broadcast name resolution traffic. The "name resolve order" parameter is of great help here. The syntax of the "name resolve order" parameter is:

```
name resolve order = wins lmhosts bcast host
```

_or_

```
name resolve order = wins lmhosts   (eliminates bcast and host)
```

The default is:

```
name  resolve order = host lmhost wins bcast
```

. where "host" refers the the native methods used by the Unix system to implement the gethostbyname() function call. This is normally controlled by `/etc/host.conf`, `/etc/nsswitch.conf` and `/etc/resolv.conf`.

Chapter 3. User information database

3.1. Introduction

Old windows clients send plain text passwords over the wire. Samba can check these passwords by crypting them and comparing them to the hash stored in the unix user database.

Newer windows clients send encrypted passwords (so-called Lanman and NT hashes) over the wire, instead of plain text passwords. The newest clients will only send encrypted passwords and refuse to send plain text passwords, unless their registry is tweaked.

These passwords can't be converted to unix style encrypted passwords. Because of that you can't use the standard unix user database, and you have to store the Lanman and NT hashes somewhere else.

Next to a differently encrypted passwords, windows also stores certain data for each user that is not stored in a unix user database, e.g. workstations the user may logon from, the location where his/her profile is stored, etc. Samba retrieves and stores this information using a "passdb backend". Commonly available backends are LDAP, plain text file, MySQL and nisplus. For more information, see the documentation about the **passdb backend =** parameter.

3.2. Important Notes About Security

The unix and SMB password encryption techniques seem similar on the surface. This similarity is, however, only skin deep. The unix scheme typically sends clear text passwords over the network when logging in. This is bad. The SMB encryption scheme never sends the cleartext password over the network but it does store the 16 byte hashed values on disk. This is also bad. Why? Because the 16 byte hashed values are a "password equivalent". You cannot derive the user's password from them, but they could potentially be used in a modified client to gain access to a server. This would require considerable technical knowledge on behalf of the attacker but is perfectly possible. You should thus treat the data stored in whatever passdb backend you use (smbpasswd file, ldap, mysql) as though it contained the cleartext passwords of all your users. Its contents must be kept secret, and the file should be protected accordingly.

Ideally we would like a password scheme which neither requires plain text passwords on the net or on disk. Unfortunately this is not available as Samba is stuck with being compatible with other SMB systems (WinNT, WfWg, Win95 etc).

Warning

Note that Windows NT 4.0 Service pack 3 changed the default for permissible authentication so that plaintext passwords are *never* sent over the wire. The solution to this is either to switch to encrypted passwords with Samba or edit the Windows NT registry to re-enable plaintext passwords. See the document WinNT.txt for details on how to do this.

Other Microsoft operating systems which also exhibit this behavior includes

These versions of MS Windows do not support full domain security protocols, although they may log onto a domain environment. Of these Only MS Windows XP Home does NOT support domain logons.

MS DOS Network client 3.0 with the basic network redirector installed
Windows 95 with the network redirector update installed
Windows 98 [se]
Windows Me
Windows XP Home

The following versions of MS Windows fully support domain security protocols.

Windows NT 3.5x
Windows NT 4.0
Windows 2000 Professional
Windows 200x Server/Advanced Server
Windows XP Professional

*Note :*All current release of Microsoft SMB/CIFS clients support authentication via the SMB Challenge/Response mechanism described here. Enabling clear text authentication does not disable the ability of the client to participate in encrypted authentication.

MS Windows clients will cache the encrypted password alone. Even when plain text passwords are re-enabled, through the appropriate registry change, the plain text password is NEVER cached. This means that in the event that a network connections should become disconnected (broken) only the cached (encrypted) password will be sent to the resource server to affect a auto-reconnect. If the resource server does not support encrypted passwords the auto-reconnect will fail. *USE OF ENCRYPTED PASSWORDS IS STRONGLY ADVISED.*

3.2.1. Advantages of SMB Encryption

Plain text passwords are not passed across the network. Someone using a network sniffer cannot just record passwords going to the SM
WinNT doesn't like talking to a server that SM not support encrypted passwords. It will refuse to browse the server if the server is also
Encrypted password support allows automatic share (resource) reconnects.

3.2.2. Advantages of non-encrypted passwords

Plain text passwords are not kept on disk, and are NOT cached in memory.
Uses same password file as other unix services such as login and ftp
Use of other services (such as telnet and ftp) which send plain text passwords over the net, so sending them for SMB isn't such a big de

3.3. The smbpasswd Command

The smbpasswd utility is a utility similar to the **passwd** or **yppasswd** programs. It maintains the two 32 byte password fields in the passdb backend.

smbpasswd works in a client-server mode where it contacts the local smbd to change the user's password on its behalf. This has enormous benefits - as follows.

smbpasswd has the capability to change passwords on Windows NT servers (this only works when the request is sent to the NT Primary Domain Controller if you are changing an NT Domain user's password).

To run smbpasswd as a normal user just type :

```
$ smbpasswd

Old SMB password: <type old value here - or hit return if there was no old
password>

New SMB Password: <type new value>

Repeat New SMB Password: <re-type new value
```

If the old value does not match the current value stored for that user, or the two new values do not match each other, then the password will not be changed.

If invoked by an ordinary user it will only allow the user to change his or her own Samba password.

If run by the root user smbpasswd may take an optional argument, specifying the user name whose SMB password you wish to change. Note that when run as root smbpasswd does not prompt for or check the old password value, thus allowing root to set passwords for users who have forgotten their passwords.

smbpasswd is designed to work in the same way and be familiar to UNIX users who use the **passwd** or **yppasswd** commands.

For more details on using **smbpasswd** refer to the man page which will always be the definitive reference.

3.4. Plain text

Older versions of samba retrieved user information from the unix user database and eventually some other fields from the file `/etc/samba/smbpasswd` or `/etc/smbpasswd`. When password encryption is disabled, no data is stored at all.

3.5. TDB

Samba can also store the user data in a "TDB" (Trivial Database). Using this backend doesn't require any additional configuration. This backend is recommended for new installations who don't require LDAP.

3.6. LDAP

3.6.1. Introduction

This document describes how to use an LDAP directory for storing Samba user account information traditionally stored in the smbpasswd(5) file. It is assumed that the reader already has a basic understanding of LDAP concepts and has a working directory server already installed. For more information on LDAP architectures and Directories, please refer to the following sites.

- OpenLDAP - http://www.openldap.org/
- iPlanet Directory Server - http://iplanet.netscape.com/directory

Note that O'Reilly Publishing (http://www.ora.com/) is working on a guide to LDAP for System Administrators which has a planned release date of early summer, 2002.

Two additional Samba resources which may prove to be helpful are

- The Samba-PDC-LDAP-HOWTO (http://www.unav.es/cti/ldap-smb/ldap-smb-3-howto.html) maintained by Ignacio Coupeau.
- The NT migration scripts from IDEALX (http://samba.idealx.org/) that are geared to manage users and group in such a Samba-LDAP Domain Controller configuration.

3.6.2. Introduction

Traditionally, when configuring "encrypt passwords = yes" (smb.conf.5.html#ENCRYPTPASSWORDS) in Samba's `smb.conf` file, user account information such as username, LM/NT password hashes, password change times, and account flags have been stored in the `smbpasswd(5)` file. There are several disadvantages to this approach for sites with very large numbers of users (counted in the thousands).

- The first is that all lookups must be performed sequentially. Given that there are approximately two lookups per domain logon (one for a normal session connection such as when mapping a network drive or printer), this is a performance bottleneck for lareg sites. What is needed is an indexed approach such as is used in databases.
- The second problem is that administrators who desired to replicate a smbpasswd file to more than one Samba server were left to use external tools such as **rsync(1)** and **ssh(1)** and wrote custom, in-house scripts.
- And finally, the amount of information which is stored in an smbpasswd entry leaves no room for additional attributes such as a home directory, password expiration time, or even a Relative Identified (RID).

As a result of these defeciencies, a more robust means of storing user attributes used by smbd was developed. The API which defines access to user accounts is commonly referred to as the samdb interface (previously this was called the passdb API, and is still so named in the CVS trees). In Samba 2.2.3, enabling support for a samdb backend (e.g. *`--with-ldapsam`* or *`--with-tdbsam`*) requires compile time support.

When compiling Samba to include the *`--with-ldapsam`* autoconf option, smbd (and associated tools) will store and lookup user accounts in an LDAP directory. In reality, this is very easy to understand. If you are comfortable with using an smbpasswd file, simply replace "smbpasswd" with "LDAP directory" in all the documentation.

There are a few points to stress about what the *`--with-ldapsam`* does not provide. The LDAP support referred to in the this documentation does not include:

- A means of retrieving user account information from an Windows 2000 Active Directory server.
- A means of replacing /etc/passwd.

The second item can be accomplished by using LDAP NSS and PAM modules. LGPL versions of these libraries can be obtained from PADL Software (http://www.padl.com/). However, the details of configuring these packages are beyond the scope of this document.

3.6.3. Supported LDAP Servers

The LDAP samdb code in 2.2.3 (and later) has been developed and tested using the OpenLDAP 2.0 server and client libraries. The same code should be able to work with Netscape's Directory Server and client SDK. However, due to lack of testing so far, there are bound to be compile errors and bugs. These should not be hard to fix. If you are so inclined, please be sure to forward all patches to samba-patches@samba.org (mailto:samba-patches@samba.org) and jerry@samba.org (mailto:jerry@samba.org).

3.6.4. Schema and Relationship to the RFC 2307 posixAccount

Samba 3.0 includes the necessary schema file for OpenLDAP 2.0 in `examples/LDAP/samba.schema`. The sambaAccount objectclass is given here:

```
objectclass ( 1.3.1.5.1.4.1.7165.2.2.2 NAME 'sambaAccount' SUP top STRUCTURAL
     DESC 'Samba Account'
     MUST ( uid $ rid )
     MAY  ( cn $ lmPassword $ ntPassword $ pwdLastSet $ logonTime $
            logoffTime $ kickoffTime $ pwdCanChange $ pwdMustChange $ acctFlags $
            displayName $ smbHome $ homeDrive $ scriptPath $ profilePath $
            description $ userWorkstations $ primaryGroupID $ domain ))
```

The samba.schema file has been formatted for OpenLDAP 2.0. The OID's are owned by the Samba Team and as such is legal to be openly published. If you translate the schema to be used with Netscape DS, please submit the modified schema file as a patch to jerry@samba.org (mailto:jerry@samba.org)

Just as the smbpasswd file is mean to store information which supplements a user's `/etc/passwd` entry, so is the sambaAccount object meant to supplement the UNIX user account information. A sambaAccount is a `STRUCTURAL` objectclass so it can be stored individually in the directory. However, there are several fields (e.g. uid) which overlap with the posixAccount objectclass outlined in RFC2307. This is by design.

In order to store all user account information (UNIX and Samba) in the directory, it is necessary to use the sambaAccount and posixAccount objectclasses in combination. However, smbd will still obtain the user's UNIX account information via the standard C library calls (e.g. getpwnam(), et. al.). This means that the Samba server must also have the LDAP NSS library installed and functioning correctly. This division of information makes it possible to store all Samba account information in LDAP, but still maintain UNIX account information in NIS while the network is transitioning to a full LDAP infrastructure.

3.6.5. Configuring Samba with LDAP

3.6.5.1. OpenLDAP configuration

To include support for the sambaAccount object in an OpenLDAP directory server, first copy the samba.schema file to slapd's configuration directory.

```
root# cp samba.schema /etc/openldap/schema/
```

Next, include the `samba.schema` file in `slapd.conf`. The sambaAccount object contains two attributes which depend upon other schema files. The 'uid' attribute is defined in `cosine.schema` and the 'displayName' attribute is defined in the `inetorgperson.schema` file. Both of these must be included before the `samba.schema` file.

```
## /etc/openldap/slapd.conf

## schema files (core.schema is required by default)
include            /etc/openldap/schema/core.schema

## needed for sambaAccount
include            /etc/openldap/schema/cosine.schema
include            /etc/openldap/schema/inetorgperson.schema
include            /etc/openldap/schema/samba.schema

## uncomment this line if you want to support the RFC2307 (NIS) schema
## include         /etc/openldap/schema/nis.schema

....
```

It is recommended that you maintain some indices on some of the most usefull attributes, like in the following example, to speed up searches made on sambaAccount objectclasses (and possibly posixAccount and posixGroup as well).

```
# Indices to maintain
## required by OpenLDAP 2.0
index objectclass   eq

## support pb_getsampwnam()
index uid           pres,eq
## support pdb_getsambapwrid()
index rid           eq

## uncomment these if you are storing posixAccount and
## posixGroup entries in the directory as well
##index uidNumber     eq
##index gidNumber     eq
##index cn            eq
##index memberUid     eq
```

3.6.5.2. Configuring Samba

The following parameters are available in smb.conf only with *`--with-ldapsam`* was included with compiling Samba.

- ldap ssl (smb.conf.5.html#LDAPSSL)
- ldap server (smb.conf.5.html#LDAPSERVER)
- ldap admin dn (smb.conf.5.html#LDAPADMINDN)
- ldap suffix (smb.conf.5.html#LDAPSUFFIX)
- ldap filter (smb.conf.5.html#LDAPFILTER)
- ldap port (smb.conf.5.html#LDAPPORT)

These are described in the smb.conf(5) (smb.conf.5.html) man page and so will not be repeated here. However, a sample smb.conf file for use with an LDAP directory could appear as

```
## /usr/local/samba/lib/smb.conf
[global]
     security = user
     encrypt passwords = yes

     netbios name = TASHTEGO
     workgroup = NARNIA

     # ldap related parameters

     # define the DN to use when binding to the directory servers
     # The password for this DN is not stored in smb.conf.  Rather it
     # must be set by using 'smbpasswd -w secretpw' to store the
     # passphrase in the secrets.tdb file.  If the "ldap admin dn" values
     # changes, this password will need to be reset.
     ldap admin dn = "cn=Samba Manager,ou=people,dc=samba,dc=org"
```

```
#  specify the LDAP server’s hostname (defaults to locahost)
ldap server = ahab.samba.org

# Define the SSL option when connecting to the directory
# (’off’, ’start tls’, or ’on’ (default))
ldap ssl = start tls

# define the port to use in the LDAP session (defaults to 636 when
# "ldap ssl = on")
ldap port = 389

# specify the base DN to use when searching the directory
ldap suffix = "ou=people,dc=samba,dc=org"

# generally the default ldap search filter is ok
# ldap filter = "(&(uid=%u)(objectclass=sambaAccount))"
```

3.6.6. Accounts and Groups management

As users accounts are managed thru the sambaAccount objectclass, you should modify you existing administration tools to deal with sambaAccount attributes.

Machines accounts are managed with the sambaAccount objectclass, just like users accounts. However, it’s up to you to stored thoses accounts in a different tree of you LDAP namespace: you should use "ou=Groups,dc=plainjoe,dc=org" to store groups and "ou=People,dc=plainjoe,dc=org" to store users. Just configure your NSS and PAM accordingly (usually, in the /etc/ldap.conf configuration file).

In Samba release 3.0, the group management system is based on posix groups. This means that Samba make usage of the posixGroup objectclass. For now, there is no NT-like group system management (global and local groups).

3.6.7. Security and sambaAccount

There are two important points to remember when discussing the security of sambaAccount entries in the directory.

- *Never* retrieve the lmPassword or ntPassword attribute values over an unencrypted LDAP session.
- *Never* allow non-admin users to view the lmPassword or ntPassword attribute values.

These password hashes are clear text equivalents and can be used to impersonate the user without deriving the original clear text strings. For more information on the details of LM/NT password hashes, refer to the ENCRYPTION chapter (ENCRYPTION.html) of the Samba-HOWTO-Collection.

To remedy the first security issue, the "ldap ssl" smb.conf parameter defaults to require an encrypted session (**ldap ssl = on**) using the default port of 636 when contacting the directory server. When using an OpenLDAP

2.0 server, it is possible to use the use the StartTLS LDAP extended operation in the place of LDAPS. In either case, you are strongly discouraged to disable this security (**ldap ssl = off**).

Note that the LDAPS protocol is deprecated in favor of the LDAPv3 StartTLS extended operation. However, the OpenLDAP library still provides support for the older method of securing communication between clients and servers.

The second security precaution is to prevent non-administrative users from harvesting password hashes from the directory. This can be done using the following ACL in `slapd.conf`:

```
## allow the "ldap admin dn" access, but deny everyone else
access to attrs=lmPassword,ntPassword
     by dn="cn=Samba Admin,ou=people,dc=plainjoe,dc=org" write
     by * none
```

3.6.8. LDAP specials attributes for sambaAccounts

The sambaAccount objectclass is composed of the following attributes:

- `lmPassword`: the LANMAN password 16-byte hash stored as a character representation of a hexidecimal string.
- `ntPassword`: the NT password hash 16-byte stored as a character representation of a hexidecimal string.
- `pwdLastSet`: The integer time in seconds since 1970 when the `lmPassword` and `ntPassword` attributes were last set.
- `acctFlags`: string of 11 characters surrounded by square brackets [] representing account flags such as U (user), W(workstation), X(no password expiration), and D(disabled).
- `logonTime`: Integer value currently unused
- `logoffTime`: Integer value currently unused
- `kickoffTime`: Integer value currently unused
- `pwdCanChange`: Integer value currently unused
- `pwdMustChange`: Integer value currently unused
- `homeDrive`: specifies the drive letter to which to map the UNC path specified by homeDirectory. The drive letter must be specified in the form "X:" where X is the letter of the drive to map. Refer to the "logon drive" parameter in the smb.conf(5) man page for more information.
- `scriptPath`: The scriptPath property specifies the path of the user's logon script, .CMD, .EXE, or .BAT file. The string can be null. The path is relative to the netlogon share. Refer to the "logon script" parameter in the smb.conf(5) man page for more information.
- `profilePath`: specifies a path to the user's profile. This value can be a null string, a local absolute path, or a UNC path. Refer to the "logon path" parameter in the smb.conf(5) man page for more information.
- `smbHome`: The homeDirectory property specifies the path of the home directory for the user. The string can be null. If homeDrive is set and specifies a drive letter, homeDirectory should be a UNC path. The path must be a

network UNC path of the form \\server\share\directory. This value can be a null string. Refer to the "logon home" parameter in the smb.conf(5) man page for more information.

- `userWorkstation`: character string value currently unused.
- `rid`: the integer representation of the user's relative identifier (RID).
- `primaryGroupID`: the relative identifier (RID) of the primary group of the user.

The majority of these parameters are only used when Samba is acting as a PDC of a domain (refer to the Samba-PDC-HOWTO (Samba-PDC-HOWTO.html) for details on how to configure Samba as a Primary Domain Controller). The following four attributes are only stored with the sambaAccount entry if the values are non-default values:

- smbHome
- scriptPath
- logonPath
- homeDrive

These attributes are only stored with the sambaAccount entry if the values are non-default values. For example, assume TASHTEGO has now been configured as a PDC and that **logon home = \\%L\%u** was defined in its `smb.conf` file. When a user named "becky" logons to the domain, the *`logon home`* string is expanded to \\TASHTEGO\becky. If the smbHome attribute exists in the entry "uid=becky,ou=people,dc=samba,dc=org", this value is used. However, if this attribute does not exist, then the value of the *`logon home`* parameter is used in its place. Samba will only write the attribute value to the directory entry is the value is something other than the default (e.g. \\MOBY\becky).

3.6.9. Example LDIF Entries for a sambaAccount

The following is a working LDIF with the inclusion of the posixAccount objectclass:

```
dn: uid=guest2, ou=people,dc=plainjoe,dc=org
ntPassword: 878D8014606CDA29677A44EFA1353FC7
pwdMustChange: 2147483647
primaryGroupID: 1201
lmPassword: 552902031BEDE9EFAAD3B435B51404EE
pwdLastSet: 1010179124
logonTime: 0
objectClass: sambaAccount
uid: guest2
kickoffTime: 2147483647
acctFlags: [UX         ]
logoffTime: 2147483647
rid: 19006
pwdCanChange: 0
```

The following is an LDIF entry for using both the sambaAccount and posixAccount objectclasses:

```
dn: uid=gcarter, ou=people,dc=plainjoe,dc=org
logonTime: 0
displayName: Gerald Carter
lmPassword: 552902031BEDE9EFAAD3B435B51404EE
primaryGroupID: 1201
objectClass: posixAccount
objectClass: sambaAccount
acctFlags: [UX         ]
userPassword: {crypt}BpM2ej8Rkzogo
uid: gcarter
uidNumber: 9000
cn: Gerald Carter
loginShell: /bin/bash
logoffTime: 2147483647
gidNumber: 100
kickoffTime: 2147483647
pwdLastSet: 1010179230
rid: 19000
homeDirectory: /home/tashtego/gcarter
pwdCanChange: 0
pwdMustChange: 2147483647
ntPassword: 878D8014606CDA29677A44EFA1353FC7
```

3.7. MySQL

3.7.1. Creating the database

You either can set up your own table and specify the field names to pdb_mysql (see below for the column names) or use the default table. The file `examples/pdb/mysql/mysql.dump` contains the correct queries to create the required tables. Use the command : **mysql -u*username* -h*hostname* -p*password* *databasename* < /path/to/samba/examples/pdb/mysql/mysql.dump**

3.7.2. Configuring

This plugin lacks some good documentation, but here is some short info:

Add a the following to the **passdb backend** variable in your `smb.conf`:

```
passdb backend = [other-plugins] mysql:identifier [other-plugins]
```

The identifier can be any string you like, as long as it doesn't collide with the identifiers of other plugins or other instances of pdb_mysql. If you specify multiple pdb_mysql.so entries in 'passdb backend', you also need to use different identifiers!

Additional options can be given thru the smb.conf file in the [global] section.

```
identifier:mysql host                     - host name, defaults to ’localhost’
identifier:mysql password
identifier:mysql user                     - defaults to ’samba’
identifier:mysql database                 - defaults to ’samba’
identifier:mysql port                     - defaults to 3306
identifier:table                          - Name of the table containing users
```

Warning

Since the password for the mysql user is stored in the smb.conf file, you should make the the smb.conf file readable only to the user that runs samba. This is considered a security bug and will be fixed soon.

Names of the columns in this table(I’ve added column types those columns should have first):

```
identifier:logon time column             - int(9)
identifier:logoff time column            - int(9)
identifier:kickoff time column           - int(9)
identifier:pass last set time column     - int(9)
identifier:pass can change time column   - int(9)
identifier:pass must change time column  - int(9)
identifier:username column               - varchar(255) - unix username
identifier:domain column                 - varchar(255) - NT domain user is part of
identifier:nt username column            - varchar(255) - NT username
identifier:fullname column            - varchar(255) - Full name of user
identifier:home dir column               - varchar(255) - Unix homedir path
identifier:dir drive column              - varchar(2) - Directory drive path (eg: ’H:’)
identifier:logon script column           - varchar(255) - Batch file to run on client side when loggi
identifier:profile path column           - varchar(255) - Path of profile
identifier:acct desc column              - varchar(255) - Some ASCII NT user data
identifier:workstations column           - varchar(255) - Workstations user can logon to (or NULL for
identifier:unknown string column         - varchar(255) - unknown string
identifier:munged dial column            - varchar(255) - ?
identifier:uid column                    - int(9) - Unix user ID (uid)
identifier:gid column                    - int(9) - Unix user group (gid)
identifier:user sid column               - varchar(255) - NT user SID
identifier:group sid column              - varchar(255) - NT group ID
identifier:lanman pass column            - varchar(255) - encrypted lanman password
identifier:nt pass column                - varchar(255) - encrypted nt passwd
identifier:plain pass column             - varchar(255) - plaintext password
identifier:acct control column           - int(9) - nt user data
identifier:unknown 3 column              - int(9) - unknown
identifier:logon divs column             - int(9) - ?
identifier:hours len column              - int(9) - ?
identifier:unknown 5 column              - int(9) - unknown
identifier:unknown 6 column              - int(9) - unknown
```

Eventually, you can put a colon (:) after the name of each column, which should specify the column to update when updating the table. You can also specify nothing behind the colon - then the data from the field will not be updated.

3.7.3. Using plaintext passwords or encrypted password

I strongly discourage the use of plaintext passwords, however, you can use them:

If you would like to use plaintext passwords, set ’identifier:lanman pass column’ and ’identifier:nt pass column’ to ’NULL’ (without the quotes) and ’identifier:plain pass column’ to the name of the column containing the plaintext passwords.

If you use encrypted passwords, set the ’identifier:plain pass column’ to ’NULL’ (without the quotes). This is the default.

3.7.4. Getting non-column data from the table

It is possible to have not all data in the database and making some ’constant’.

For example, you can set ’identifier:fullname column’ to : **CONCAT(First_name,’ ’,Sur_name)**

Or, set ’identifier:workstations column’ to : **NULL**

See the MySQL documentation for more language constructs.

3.8. XML

This module requires libxml2 to be installed.

The usage of pdb_xml is pretty straightforward. To export data, use: `pdbedit -e xml:filename` (where filename is the name of the file to put the data in)

To import data, use: `pdbedit -i xml:filename -e current-pdb` Where filename is the name to read the data from and current-pdb to put it in.

II. Type of installation

Introduction

Samba can operate in various SMB networks. This part contains information on configuring samba for various environments.

Chapter 4. Nomenclature of Server Types

Adminstrators of Microsoft networks often refer to there being three different type of servers:

- Stand Alone Server
- Domain Member Server
- Domain Controller
  - Primary Domain Controller
  - Backup Domain Controller
  - ADS Domain Controller

A network administrator who is familiar with these terms and who wishes to migrate to or use Samba will want to know what these terms mean within a Samba context.

4.1. Stand Alone Server

The term *stand alone server* means that the server will provide local authentication and access control for all resources that are available from it. In general this means that there will be a local user database. In more technical terms, it means that resources on the machine will either be made available in either SHARE mode or in USER mode. SHARE mode and USER mode security are documented under discussions regarding "security mode". The smb.conf configuration parameters that control security mode are: "security = user" and "security = share".

No special action is needed other than to create user accounts. Stand-alone servers do NOT provide network logon services, meaning that machines that use this server do NOT perform a domain logon but instead make use only of the MS Windows logon which is local to the MS Windows workstation/server.

Samba tends to blur the distinction a little in respect of what is a stand alone server. This is because the authentication database may be local or on a remote server, even if from the samba protocol perspective the samba server is NOT a member of a domain security context.

Through the use of PAM (Pluggable Authentication Modules) and nsswitch (the name service switcher) the source of authentication may reside on another server. We would be inclined to call this the authentication server. This means that the samba server may use the local Unix/Linux system password database (/etc/passwd or /etc/shadow), may use a local smbpasswd file (/etc/samba/smbpasswd or /usr/local/samba/lib/private/smbpasswd), or may use an LDAP back end, or even via PAM and Winbind another CIFS/SMB server for authentication.

4.2. Domain Member Server

This mode of server operation involves the samba machine being made a member of a domain security context. This means by definition that all user authentication will be done from a centrally defined authentication regime. The authentication regime may come from an NT3/4 style (old domain technology) server, or it may be provided from an Active Directory server (ADS) running on MS Windows 2000 or later.

Of course it should be clear that the authentication back end itself could be from any distributed directory architecture server that is supported by Samba. This can be LDAP (from OpenLDAP), or Sun's iPlanet, of NetWare Directory Server, etc.

Please refer to the section on Howto configure Samba as a Primary Domain Controller and for more information regarding how to create a domain machine account for a domain member server as well as for information regading how to enable the samba domain member machine to join the domain and to be fully trusted by it.

4.3. Domain Controller

Over the years public perceptions of what Domain Control really is has taken on an almost mystical nature. Before we branch into a brief overview of what Domain Control is the following types of controller are known:

4.3.1. Domain Controller Types

Primary Domain Controller
Backup Domain Controller
ADS Domain Controller

The *Primary Domain Controller* or PDC plays an important role in the MS Windows NT3 and NT4 Domain Control architecture, but not in the manner that so many expect. The PDC seeds the Domain Control database (a part of the Windows registry) and it plays a key part in synchronisation of the domain authentication database.

New to Samba-3.0.0 is the ability to use a back-end file that holds the same type of data as the NT4 style SAM (Security Account Manager) database (one of the registry files). The samba-3.0.0 SAM can be specified via the smb.conf file parameter "passwd backend" and valid options include *smbpasswd tdbsam ldapsam nisplussam plugin unixsam*. The smbpasswd, tdbsam and ldapsam options can have a "_nua" suffix to indicate that No Unix Accounts need to be created. In other words, the Samba SAM will be independant of Unix/Linux system accounts, provided a uid range is defined from which SAM accounts can be created.

The *Backup Domain Controller* or BDC plays a key role in servicing network authentication requests. The BDC is biased to answer logon requests so that on a network segment that has a BDC and a PDC the BDC will be most likely to service network logon requests. The PDC will answer network logon requests when the BDC is too busy (high load). A BDC can be promoted to a PDC. If the PDC is on line at the time that the BDC is promoted to PDC the previous PDC is automatically demoted to a BDC.

At this time Samba is NOT capable of acting as an *ADS Domain Controller*.

Chapter 5. Samba as Stand-Alone Server

In this section the function and purpose of Samba's *security* modes are described.

5.1. User and Share security level

A SMB server tells the client at startup what "security level" it is running. There are two options "share level" and "user level". Which of these two the client receives affects the way the client then tries to authenticate itself. It does not directly affect (to any great extent) the way the Samba server does security. I know this is strange, but it fits in with the client/server approach of SMB. In SMB everything is initiated and controlled by the client, and the server can only tell the client what is available and whether an action is allowed.

5.1.1. User Level Security

I'll describe user level security first, as its simpler. In user level security the client will send a "session setup" command directly after the protocol negotiation. This contains a username and password. The server can either accept or reject that username/password combination. Note that at this stage the server has no idea what share the client will eventually try to connect to, so it can't base the "accept/reject" on anything other than:

1. the username/password
2. the machine that the client is coming from

If the server accepts the username/password then the client expects to be able to mount any share (using a "tree connection") without specifying a password. It expects that all access rights will be as the username/password specified in the "session setup".

It is also possible for a client to send multiple "session setup" requests. When the server responds it gives the client a "uid" to use as an authentication tag for that username/password. The client can maintain multiple authentication contexts in this way (WinDD is an example of an application that does this)

5.1.2. Share Level Security

Ok, now for share level security. In share level security the client authenticates itself separately for each share. It will send a password along with each "tree connection" (share mount). It does not explicitly send a username with this operation. The client is expecting a password to be associated with each share, independent of the user. This means that samba has to work out what username the client probably wants to use. It is never explicitly sent the username. Some commercial SMB servers such as NT actually associate passwords directly with shares in share level security, but samba always uses the unix authentication scheme where it is a username/password that is authenticated, not a "share/password".

Many clients send a "session setup" even if the server is in share level security. They normally send a valid username but no password. Samba records this username in a list of "possible usernames". When the client then does a "tree connection" it also adds to this list the name of the share they try to connect to (useful for home directories) and any users listed in the **user =** `smb.conf` line. The password is then checked in turn against these "possible usernames". If a match is found then the client is authenticated as that user.

5.1.3. Server Level Security

Finally "server level" security. In server level security the samba server reports to the client that it is in user level security. The client then does a "session setup" as described earlier. The samba server takes the username/password that the client sends and attempts to login to the "password server" by sending exactly the same username/password that it got from the client. If that server is in user level security and accepts the password then samba accepts the clients connection. This allows the samba server to use another SMB server as the "password server".

You should also note that at the very start of all this, where the server tells the client what security level it is in, it also tells the client if it supports encryption. If it does then it supplies the client with a random "cryptkey". The client will then send all passwords in encrypted form. You have to compile samba with encryption enabled to support this feature, and you have to maintain a separate smbpasswd file with SMB style encrypted passwords. It is cryptographically impossible to translate from unix style encryption to SMB style encryption, although there are some fairly simple management schemes by which the two could be kept in sync.

"security = server" means that Samba reports to clients that it is running in "user mode" but actually passes off all authentication requests to another "user mode" server. This requires an additional parameter "password server =" that points to the real authentication server. That real authentication server can be another Samba server or can be a Windows NT server, the later natively capable of encrypted password support.

5.1.3.1. Configuring Samba for Seemless Windows Network Integration

MS Windows clients may use encrypted passwords as part of a challenege/response authentication model (a.k.a. NTLMv1) or alone, or clear text strings for simple password based authentication. It should be realized that with the SMB protocol the password is passed over the network either in plain text or encrypted, but not both in the same authentication requests.

When encrypted passwords are used a password that has been entered by the user is encrypted in two ways:

- An MD4 hash of the UNICODE of the password string. This is known as the NT hash.
- The password is converted to upper case, and then padded or trucated to 14 bytes. This string is then appended with 5 bytes of NULL characters and split to form two 56 bit DES keys to encrypt a "magic" 8 byte value. The resulting 16 bytes for the LanMan hash.

MS Windows 95 pre-service pack 1, MS Windows NT versions 3.x and version 4.0 pre-service pack 3 will use either mode of password authentication. All versions of MS Windows that follow these versions no longer support plain text passwords by default.

MS Windows clients have a habit of dropping network mappings that have been idle for 10 minutes or longer. When the user attempts to use the mapped drive connection that has been dropped, the client re-establishes the connection using a cached copy of the password.

When Microsoft changed the default password mode, support was dropped for caching of the plain text password. This means that when the registry parameter is changed to re-enable use of plain text passwords it appears to work, but when a dropped service connection mapping attempts to revalidate it will fail if the remote authentication server does not support encrypted passwords. This means that it is definitely not a good idea to re-enable plain text password support in such clients.

The following parameters can be used to work around the issue of Windows 9x client upper casing usernames and password before transmitting them to the SMB server when using clear text authentication.

```
passsword level (smb.conf.5.html#PASSWORDLEVEL) = integer
username level (smb.conf.5.html#USERNAMELEVEL) = integer
```

By default Samba will lower case the username before attempting to lookup the user in the database of local system accounts. Because UNIX usernames conventionally only contain lower case character, the *`username level`* parameter is rarely needed.

However, passwords on UNIX systems often make use of mixed case characters. This means that in order for a user on a Windows 9x client to connect to a Samba server using clear text authentication, the *`password level`* must be set to the maximum number of upper case letter which *could* appear is a password. Note that is the server OS uses the traditional DES version of crypt(), then a *`password level`* of 8 will result in case insensitive passwords as seen from Windows users. This will also result in longer login times as Samba hash to compute the permutations of the password string and try them one by one until a match is located (or all combinations fail).

The best option to adopt is to enable support for encrypted passwords where ever Samba is used. There are three configuration possibilities for support of encrypted passwords:

5.1.3.2. Use MS Windows NT as an authentication server

This method involves the additions of the following parameters in the `smb.conf` file:

```
encrypt passwords = Yes
security = server
password server = "NetBIOS_name_of_PDC"
```

There are two ways of identifying whether or not a username and password pair was valid or not. One uses the reply information provided as part of the authentication messaging process, the other uses just and error code.

The down-side of this mode of configuration is the fact that for security reasons Samba will send the password server a bogus username and a bogus password and if the remote server fails to reject the username and password pair then an alternative mode of identification of validation is used. Where a site uses password lock out after a certain number of failed authentication attempts this will result in user lockouts.

Use of this mode of authentication does require there to be a standard Unix account for the user, this account can be blocked to prevent logons by other than MS Windows clients.

5.1.4. Domain Level Security

When samba is operating in *security = domain* mode this means that the Samba server has a domain security trust account (a machine account) and will cause all authentication requests to be passed through to the domain controllers.

5.1.4.1. Samba as a member of an MS Windows NT security domain

This method involves additon of the following paramters in the `smb.conf` file:

```
encrypt passwords = Yes
security = domain
workgroup = "name of NT domain"
password server = *
```

The use of the "*" argument to **password server** will cause samba to locate the domain controller in a way analogous to the way this is done within MS Windows NT. This is the default behaviour.

In order for this method to work the Samba server needs to join the MS Windows NT security domain. This is done as follows:

- On the MS Windows NT domain controller using the Server Manager add a machine account for the Samba server.
- Next, on the Linux system execute: **smbpasswd -r PDC_NAME -j DOMAIN_NAME**

Use of this mode of authentication does require there to be a standard Unix account for the user in order to assign a uid once the account has been authenticated by the remote Windows DC. This account can be blocked to prevent logons by other than MS Windows clients by things such as setting an invalid shell in the `/etc/passwd` entry.

An alternative to assigning UIDs to Windows users on a Samba member server is presented in the Winbind Overview (winbind.html) chapter in this HOWTO collection.

5.1.5. ADS Level Security

For information about the configuration option please refer to the entire section entitled *Samba as an ADS Domain Member*.

Chapter 6. Samba as an NT4 or Win2k Primary Domain Controller

6.1. Prerequisite Reading

Before you continue reading in this chapter, please make sure that you are comfortable with configuring basic files services in smb.conf and how to enable and administer password encryption in Samba. Theses two topics are covered in the `smb.conf(5)` (smb.conf.5.html) manpage.

6.2. Background

This article outlines the steps necessary for configuring Samba as a PDC. It is necessary to have a working Samba server prior to implementing the PDC functionality.

- Domain logons for Windows NT 4.0 / 200x / XP Professional clients.
- Placing Windows 9x / Me clients in user level security
- Retrieving a list of users and groups from a Samba PDC to Windows 9x / Me / NT / 200x / XP Professional clients
- Roaming Profiles
- Network/System Policies

> **Note:** Roaming Profiles and System/Network policies are advanced network administration topics that are covered separately in this document.

The following functionalities are new to the Samba 3.0 release:

- Windows NT 4 domain trusts
- Adding users via the User Manager for Domains

The following functionalities are NOT provided by Samba 3.0:

- SAM replication with Windows NT 4.0 Domain Controllers (i.e. a Samba PDC and a Windows NT BDC or vice versa)
- Acting as a Windows 2000 Domain Controller (i.e. Kerberos and Active Directory)

Please note that Windows 9x / Me / XP Home clients are not true members of a domain for reasons outlined in this article. Therefore the protocol for support Windows 9x-style domain logons is completely different from NT4 / Win2k type domain logons and has been officially supported for some time.

MS Windows XP Home edition is NOT able to join a domain and does not permit the use of domain logons.

Implementing a Samba PDC can basically be divided into 3 broad steps.

1. Configuring the Samba PDC
2. Creating machine trust accounts and joining clients to the domain
3. Adding and managing domain user accounts

There are other minor details such as user profiles, system policies, etc... However, these are not necessarily specific to a Samba PDC as much as they are related to Windows NT networking concepts.

6.3. Configuring the Samba Domain Controller

The first step in creating a working Samba PDC is to understand the parameters necessary in smb.conf. Here we attempt to explain the parameters that are covered in the smb.conf man page (smb.conf.5.html).

Here is an example `smb.conf` for acting as a PDC:

```
[global]
    ; Basic server settings
    netbios name (smb.conf.5.html#NETBIOSNAME) = POGO
    workgroup (smb.conf.5.html#WORKGROUP) = NARNIA

    ; we should act as the domain and local master browser
    os level (smb.conf.5.html#OSLEVEL) = 64
    preferred master (smb.conf.5.html#PERFERREDMASTER) = yes
    domain master (smb.conf.5.html#DOMAINMASTER) = yes
    local master (smb.conf.5.html#LOCALMASTER) = yes

    ; security settings (must user security = user)
    security (smb.conf.5.html#SECURITYEQUALSUSER) = user

    ; encrypted passwords are a requirement for a PDC
    encrypt passwords (smb.conf.5.html#ENCRYPTPASSWORDS) = yes

    ; support domain logons
    domain logons (smb.conf.5.html#DOMAINLOGONS) = yes

    ; where to store user profiles?
    logon path (smb.conf.5.html#LOGONPATH) = \\%N\profiles\%u

    ; where is a user's home directory and where should it be mounted at?
    logon drive (smb.conf.5.html#LOGONDRIVE) = H:
    logon home (smb.conf.5.html#LOGONHOME) = \\homeserver\%u

    ; specify a generic logon script for all users
    ; this is a relative **DOS** path to the [netlogon] share
    logon script (smb.conf.5.html#LOGONSCRIPT) = logon.cmd

; necessary share for domain controller
[netlogon]
    path (smb.conf.5.html#PATH) = /usr/local/samba/lib/netlogon
```

```
    read only (smb.conf.5.html#READONLY) = yes
    write list (smb.conf.5.html#WRITELIST) = ntadmin

; share for storing user profiles
[profiles]
    path (smb.conf.5.html#PATH) = /export/smb/ntprofile
    read only (smb.conf.5.html#READONLY) = no
    create mask (smb.conf.5.html#CREATEMASK) = 0600
    directory mask (smb.conf.5.html#DIRECTORYMASK) = 0700
```

There are a couple of points to emphasize in the above configuration.

- Encrypted passwords must be enabled. For more details on how to do this, refer to ENCRYPTION.html.
- The server must support domain logons and a `[netlogon]` share
- The server must be the domain master browser in order for Windows client to locate the server as a DC. Please refer to the various Network Browsing documentation included with this distribution for details.

Samba 3.0 offers a complete implementation of group mapping between Windows NT groups and Unix groups (this is really quite complicated to explain in a short space).

6.4. Creating Machine Trust Accounts and Joining Clients to the Domain

A machine trust account is a Samba account that is used to authenticate a client machine (rather than a user) to the Samba server. In Windows terminology, this is known as a "Computer Account."

The password of a machine trust account acts as the shared secret for secure communication with the Domain Controller. This is a security feature to prevent an unauthorized machine with the same NetBIOS name from joining the domain and gaining access to domain user/group accounts. Windows NT, 200x, XP Professional clients use machine trust accounts, but Windows 9x / Me / XP Home clients do not. Hence, a Windows 9x / Me / XP Home client is never a true member of a domain because it does not possess a machine trust account, and thus has no shared secret with the domain controller.

A Windows PDC stores each machine trust account in the Windows Registry. A Samba-3 PDC also has to stoe machine trust account information in a suitable back-end data store. With Samba-3 there can be multiple back-ends for this including:

- *smbpaswd* - the plain ascii file stored used by earlier versions of Samba. This file configuration option requires a Unix/Linux system account for EVERY entry (ie: both for user and for machine accounts). This file will be located in the *private* directory (default is /usr/local/samba/lib/private or on linux /etc/samba).
- *smbpasswd_nua* - This file is independant of the system wide user accounts. The use of this back-end option requires specification of the "non unix account range" option also. It is called smbpasswd and will be located in the `private` directory.

- *tdbsam* - a binary database backend that will be stored in the *private* directory in a file called *passwd.tdb*. The key benefit of this binary format file is that it can store binary objects that can not be accomodated in the traditional plain text smbpasswd file.
- *tdbsam_nua* like the smbpasswd_nua option above, this file allows the creation of arbitrary user and machine accounts without requiring that account to be added to the system (/etc/passwd) file. It too requires the specification of the "non unix account range" option in the [globals] section of the smb.conf file.
- *ldapsam* - An LDAP based back-end. Permits the LDAP server to be specified. eg: ldap://localhost or ldap://frodo.murphy.com
- *ldapsam_nua* - LDAP based back-end with no unix account requirement, like smbpasswd_nua and tdbsam_nua above.

A Samba PDC, however, stores each machine trust account in two parts, as follows:

- A Samba account, stored in the same location as user LanMan and NT password hashes (currently `smbpasswd`). The Samba account possesses and uses only the NT password hash.
- A corresponding Unix account, typically stored in `/etc/passwd`. (Future releases will alleviate the need to create `/etc/passwd` entries.)

There are two ways to create machine trust accounts:

- Manual creation. Both the Samba and corresponding Unix account are created by hand.
- "On-the-fly" creation. The Samba machine trust account is automatically created by Samba at the time the client is joined to the domain. (For security, this is the recommended method.) The corresponding Unix account may be created automatically or manually.

6.4.1. Manual Creation of Machine Trust Accounts

The first step in manually creating a machine trust account is to manually create the corresponding Unix account in `/etc/passwd`. This can be done using **vipw** or other 'add user' command that is normally used to create new Unix accounts. The following is an example for a Linux based Samba server:

```
root# /usr/sbin/useradd -g 100 -d /dev/null -c "machine nickname" -s /bin/false machine_name$
```

```
root# passwd -l machine_name$
```

On *BSD systems, this can be done using the 'chpass' utility:

```
root# chpass -a "machine_name$:*:101:100::0:0:Workstation
machine_name:/dev/null:/sbin/nologin"
```

The `/etc/passwd` entry will list the machine name with a "$" appended, won't have a password, will have a null shell and no home directory. For example a machine named 'doppy' would have an `/etc/passwd` entry like this:

```
doppy$:x:505:501:machine_nickname:/dev/null:/bin/false
```

Above, `machine_nickname` can be any descriptive name for the client, i.e., BasementComputer. `machine_name` absolutely must be the NetBIOS name of the client to be joined to the domain. The "$" must be appended to the NetBIOS name of the client or Samba will not recognize this as a machine trust account.

Now that the corresponding Unix account has been created, the next step is to create the Samba account for the client containing the well-known initial machine trust account password. This can be done using the **smbpasswd(8)** (smbpasswd.8.html) command as shown here:

```
root# smbpasswd -a -m machine_name
```

where `machine_name` is the machine's NetBIOS name. The RID of the new machine account is generated from the UID of the corresponding Unix account.

Join the client to the domain immediately

Manually creating a machine trust account using this method is the equivalent of creating a machine trust account on a Windows NT PDC using the "Server Manager". From the time at which the account is created to the time which the client joins the domain and changes the password, your domain is vulnerable to an intruder joining your domain using a a machine with the same NetBIOS name. A PDC inherently trusts members of the domain and will serve out a large degree of user information to such clients. You have been warned!

6.4.2. "On-the-Fly" Creation of Machine Trust Accounts

The second (and recommended) way of creating machine trust accounts is simply to allow the Samba server to create them as needed when the client is joined to the domain.

Since each Samba machine trust account requires a corresponding Unix account, a method for automatically creating the Unix account is usually supplied; this requires configuration of the add user script (smb.conf.5.html#ADDUSERSCRIPT) option in `smb.conf`. This method is not required, however; corresponding Unix accounts may also be created manually.

Below is an example for a RedHat 6.2 Linux system.

```
[global]
   # <...remainder of parameters...>
   add user script = /usr/sbin/useradd -d /dev/null -g 100 -s /bin/false -M %u
```

6.4.3. Joining the Client to the Domain

The procedure for joining a client to the domain varies with the version of Windows.

- *Windows 2000*

  When the user elects to join the client to a domain, Windows prompts for an account and password that is privileged to join the domain. A Samba administrative account (i.e., a Samba account that has root privileges on the Samba server) must be entered here; the operation will fail if an ordinary user account is given. The password for this account should be set to a different password than the associated `/etc/passwd` entry, for security reasons.

  The session key of the Samba administrative account acts as an encryption key for setting the password of the machine trust account. The machine trust account will be created on-the-fly, or updated if it already exists.

- *Windows NT*

  If the machine trust account was created manually, on the Identification Changes menu enter the domain name, but do not check the box "Create a Computer Account in the Domain." In this case, the existing machine trust account is used to join the machine to the domain.

  If the machine trust account is to be created on-the-fly, on the Identification Changes menu enter the domain name, and check the box "Create a Computer Account in the Domain." In this case, joining the domain proceeds as above for Windows 2000 (i.e., you must supply a Samba administrative account when prompted).

6.5. Common Problems and Errors

- *I cannot include a '$' in a machine name.*

  A 'machine name' in (typically) `/etc/passwd` of the machine name with a '$' appended. FreeBSD (and other BSD systems?) won't create a user with a '$' in their name.

  The problem is only in the program used to make the entry, once made, it works perfectly. So create a user without the '$' and use **vipw** to edit the entry, adding the '$'. Or create the whole entry with vipw if you like, make sure you use a unique User ID !

- *I get told "You already have a connection to the Domain...." or "Cannot join domain, the credentials supplied conflict with an existing set.." when creating a machine trust account.*

  This happens if you try to create a machine trust account from the machine itself and already have a connection (e.g. mapped drive) to a share (or IPC$) on the Samba PDC. The following command will remove all network drive connections:

```
C:\WINNT\> net use * /d
```

Further, if the machine is a already a 'member of a workgroup' that is the same name as the domain you are joining (bad idea) you will get this message. Change the workgroup name to something else, it does not matter what, reboot, and try again.

- *The system can not log you on (C000019B)....*

  I joined the domain successfully but after upgrading to a newer version of the Samba code I get the message, "The system can not log you on (C000019B), Please try again or consult your system administrator" when attempting to logon.

  This occurs when the domain SID stored in the secrets.tdb database is changed. The most common cause of a change in domain SID is when the domain name and/or the server name (netbios name) is changed. The only way to correct the problem is to restore the original domain SID or remove the domain client from the domain and rejoin. The domain SID may be reset using either the smbpasswd or rpcclient utilities.

- *The machine trust account for this computer either does not exist or is not accessible.*

  When I try to join the domain I get the message "The machine account for this computer either does not exist or is not accessible". What's wrong?

  This problem is caused by the PDC not having a suitable machine trust account. If you are using the `add user script` method to create accounts then this would indicate that it has not worked. Ensure the domain admin user system is working.

  Alternatively if you are creating account entries manually then they have not been created correctly. Make sure that you have the entry correct for the machine trust account in smbpasswd file on the Samba PDC. If you added the account using an editor rather than using the smbpasswd utility, make sure that the account name is the machine NetBIOS name with a '$' appended to it (i.e. computer_name$). There must be an entry in both /etc/passwd and the smbpasswd file. Some people have reported that inconsistent subnet masks between the Samba server and the NT client have caused this problem. Make sure that these are consistent for both client and server.

- *When I attempt to login to a Samba Domain from a NT4/W2K workstation, I get a message about my account being disabled.*

  This problem is caused by a PAM related bug in Samba 2.2.0. This bug is fixed in 2.2.1. Other symptoms could be unaccessible shares on NT/W2K member servers in the domain or the following error in your smbd.log: passdb/pampass.c:pam_account(268) PAM: UNKNOWN ERROR for User: %user%

  At first be ensure to enable the useraccounts with **smbpasswd -e %user%**, this is normally done, when you create an account.

In order to work around this problem in 2.2.0, configure the *`account`* control flag in `/etc/pam.d/samba` file as follows:

```
account required        pam_permit.so
```

If you want to remain backward compatibility to samba 2.0.x use `pam_permit.so`, it's also possible to use `pam_pwdb.so`. There are some bugs if you try to use `pam_unix.so`, if you need this, be ensure to use the most recent version of this file.

6.6. What other help can I get?

There are many sources of information available in the form of mailing lists, RFC's and documentation. The docs that come with the samba distribution contain very good explanations of general SMB topics such as browsing.

- *What are some diagnostics tools I can use to debug the domain logon process and where can I find them?*

  One of the best diagnostic tools for debugging problems is Samba itself. You can use the -d option for both smbd and nmbd to specify what 'debug level' at which to run. See the man pages on smbd, nmbd and smb.conf for more information on debugging options. The debug level can range from 1 (the default) to 10 (100 for debugging passwords).

  Another helpful method of debugging is to compile samba using the **gcc -g** flag. This will include debug information in the binaries and allow you to attach gdb to the running smbd / nmbd process. In order to attach gdb to an smbd process for an NT workstation, first get the workstation to make the connection. Pressing ctrl-alt-delete and going down to the domain box is sufficient (at least, on the first time you join the domain) to generate a 'LsaEnumTrustedDomains'. Thereafter, the workstation maintains an open connection, and therefore there will be an smbd process running (assuming that you haven't set a really short smbd idle timeout) So, in between pressing ctrl alt delete, and actually typing in your password, you can gdb attach and continue.

  Some useful samba commands worth investigating:

  - testparam | more
  - smbclient -L //{netbios name of server}

  An SMB enabled version of tcpdump is available from http://www.tcpdup.org/ (http://www.tcpdump.org/). Ethereal, another good packet sniffer for Unix and Win32 hosts, can be downloaded from http://www.ethereal.com (http://www.ethereal.com/).

  For tracing things on the Microsoft Windows NT, Network Monitor (aka. netmon) is available on the Microsoft Developer Network CD's, the Windows NT Server install CD and the SMS CD's. The version of netmon that ships with SMS allows for dumping packets between any two computers (i.e. placing the network interface in promiscuous mode). The version on the NT Server install CD will only allow monitoring of network traffic directed to the local NT box and broadcasts on the local subnet. Be aware that Ethereal can read and write netmon formatted files.

- *How do I install 'Network Monitor' on an NT Workstation or a Windows 9x box?*

  Installing netmon on an NT workstation requires a couple of steps. The following are for installing Netmon V4.00.349, which comes with Microsoft Windows NT Server 4.0, on Microsoft Windows NT Workstation 4.0. The process should be similar for other version of Windows NT / Netmon. You will need both the Microsoft Windows NT Server 4.0 Install CD and the Workstation 4.0 Install CD.

  Initially you will need to install 'Network Monitor Tools and Agent' on the NT Server. To do this

  - Goto Start - Settings - Control Panel - Network - Services - Add
  - Select the 'Network Monitor Tools and Agent' and click on 'OK'.
  - Click 'OK' on the Network Control Panel.
  - Insert the Windows NT Server 4.0 install CD when prompted.

  At this point the Netmon files should exist in `%SYSTEMROOT%\System32\netmon\*.*`. Two subdirectories exist as well, `parsers\` which contains the necessary DLL's for parsing the netmon packet dump, and `captures\`.

  In order to install the Netmon tools on an NT Workstation, you will first need to install the 'Network Monitor Agent' from the Workstation install CD.

  - Goto Start - Settings - Control Panel - Network - Services - Add
  - Select the 'Network Monitor Agent' and click on 'OK'.
  - Click 'OK' on the Network Control Panel.
  - Insert the Windows NT Workstation 4.0 install CD when prompted.

  Now copy the files from the NT Server in %SYSTEMROOT%\System32\netmon*.* to %SYSTEMROOT%\System32\netmon*.* on the Workstation and set permissions as you deem appropriate for your site. You will need administrative rights on the NT box to run netmon.

  To install Netmon on a Windows 9x box install the network monitor agent from the Windows 9x CD (\admin\nettools\netmon). There is a readme file located with the netmon driver files on the CD if you need information on how to do this. Copy the files from a working Netmon installation.

- The following is a list if helpful URLs and other links:
  - Home of Samba site http://samba.org (http://samba.org). We have a mirror near you !
  - The *Development* document on the Samba mirrors might mention your problem. If so, it might mean that the developers are working on it.
  - See how Scott Merrill simulates a BDC behavior at http://www.skippy.net/linux/smb-howto.html (http://www.skippy.net/linux/smb-howto.html).

  - Although 2.0.7 has almost had its day as a PDC, David Bannon will keep the 2.0.7 PDC pages at http://bioserve.latrobe.edu.au/samba (http://bioserve.latrobe.edu.au/samba) going for a while yet.
  - Misc links to CIFS information http://samba.org/cifs/
  - NT Domains for Unix http://mailhost.cb1.com/~lkcl/ntdom/ (http://mailhost.cb1.com/~lkcl/ntdom/)
  - FTP site for older SMB specs: ftp://ftp.microsoft.com/developr/drg/CIFS/ (ftp://ftp.microsoft.com/developr/drg/CIFS/)

- *How do I get help from the mailing lists?*

  There are a number of Samba related mailing lists. Go to http://samba.org, click on your nearest mirror and then click on **Support** and then click on **Samba related mailing lists**.

  For questions relating to Samba TNG go to http://www.samba-tng.org/ It has been requested that you don't post questions about Samba-TNG to the main stream Samba lists.

  If you post a message to one of the lists please observe the following guide lines :

  - Always remember that the developers are volunteers, they are not paid and they never guarantee to produce a particular feature at a particular time. Any time lines are 'best guess' and nothing more.
  - Always mention what version of samba you are using and what operating system its running under. You should probably list the relevant sections of your smb.conf file, at least the options in [global] that affect PDC support.
  - In addition to the version, if you obtained Samba via CVS mention the date when you last checked it out.
  - Try and make your question clear and brief, lots of long, convoluted questions get deleted before they are completely read ! Don't post html encoded messages (if you can select colour or font size its html).
  - If you run one of those nifty 'I'm on holidays' things when you are away, make sure its configured to not answer mailing lists.
  - Don't cross post. Work out which is the best list to post to and see what happens, i.e. don't post to both samba-ntdom and samba-technical. Many people active on the lists subscribe to more than one list and get annoyed to see the same message two or more times. Often someone will see a message and thinking it would be better dealt with on another, will forward it on for you.
  - You might include *partial* log files written at a debug level set to as much as 20. Please don't send the entire log but enough to give the context of the error messages.
  - (Possibly) If you have a complete netmon trace (from the opening of the pipe to the error) you can send the *.CAP file as well.
  - Please think carefully before attaching a document to an email. Consider pasting the relevant parts into the body of the message. The samba mailing lists go to a huge number of people, do they all need a copy of your smb.conf in their attach directory?

- *How do I get off the mailing lists?*

  To have your name removed from a samba mailing list, go to the same place you went to to get on it. Go to http://lists.samba.org (http://lists.samba.org/), click on your nearest mirror and then click on **Support** and then click on **Samba related mailing lists**. Or perhaps see here (http://lists.samba.org/mailman/roster/samba-ntdom)

Please don't post messages to the list asking to be removed, you will just be referred to the above address (unless that process failed in some way...)

6.7. Domain Control for Windows 9x/ME

A domain and a workgroup are exactly the same thing in terms of network browsing. The difference is that a distributable authentication database is associated with a domain, for secure login access to a network. Also, different access rights can be granted to users if they successfully authenticate against a domain logon server. Samba-3 does this now in the same way that MS Windows NT/2K.

The SMB client logging on to a domain has an expectation that every other server in the domain should accept the same authentication information. Network browsing functionality of domains and workgroups is identical and is explained in this documentation under the browsing discussions. It should be noted, that browsing is totally orthogonal to logon support.

Issues related to the single-logon network model are discussed in this section. Samba supports domain logons, network logon scripts, and user profiles for MS Windows for workgroups and MS Windows 9X/ME clients which are the focus of this section.

When an SMB client in a domain wishes to logon it broadcast requests for a logon server. The first one to reply gets the job, and validates its password using whatever mechanism the Samba administrator has installed. It is possible (but very stupid) to create a domain where the user database is not shared between servers, i.e. they are effectively workgroup servers advertising themselves as participating in a domain. This demonstrates how authentication is quite different from but closely involved with domains.

Using these features you can make your clients verify their logon via the Samba server; make clients run a batch file when they logon to the network and download their preferences, desktop and start menu.

Before launching into the configuration instructions, it is worthwhile lookingat how a Windows 9x/ME client performs a logon:

1. The client broadcasts (to the IP broadcast address of the subnet it is in) a NetLogon request. This is sent to the NetBIOS name DOMAIN<1c> at the NetBIOS layer. The client chooses the first response it receives, which contains the NetBIOS name of the logon server to use in the format of \\SERVER.
2. The client then connects to that server, logs on (does an SMBsessetupX) and then connects to the IPC$ share (using an SMBtconX).
3. The client then does a NetWkstaUserLogon request, which retrieves the name of the user's logon script.
4. The client then connects to the NetLogon share and searches for this and if it is found and can be read, is retrieved and executed by the client. After this, the client disconnects from the NetLogon share.

5. The client then sends a NetUserGetInfo request to the server, to retrieve the user's home share, which is used to search for profiles. Since the response to the NetUserGetInfo request does not contain much more the user's home share, profiles for Win9X clients MUST reside in the user home directory.
6. The client then connects to the user's home share and searches for the user's profile. As it turns out, you can specify the user's home share as a sharename and path. For example, \\server\fred\.profile. If the profiles are found, they are implemented.
7. The client then disconnects from the user's home share, and reconnects to the NetLogon share and looks for CONFIG.POL, the policies file. If this is found, it is read and implemented.

6.7.1. Configuration Instructions: Network Logons

The main difference between a PDC and a Windows 9x logon server configuration is that

- Password encryption is not required for a Windows 9x logon server.
- Windows 9x/ME clients do not possess machine trust accounts.

Therefore, a Samba PDC will also act as a Windows 9x logon server.

security mode and master browsers

There are a few comments to make in order to tie up some loose ends. There has been much debate over the issue of whether or not it is ok to configure Samba as a Domain Controller in security modes other than `USER`. The only security mode which will not work due to technical reasons is `SHARE` mode security. `DOMAIN` and `SERVER` mode security is really just a variation on SMB user level security.

Actually, this issue is also closely tied to the debate on whether or not Samba must be the domain master browser for its workgroup when operating as a DC. While it may technically be possible to configure a server as such (after all, browsing and domain logons are two distinctly different functions), it is not a good idea to so. You should remember that the DC must register the DOMAIN#1b NetBIOS name. This is the name used by Windows clients to locate the DC. Windows clients do not distinguish between the DC and the DMB. For this reason, it is very wise to configure the Samba DC as the DMB.

Now back to the issue of configuring a Samba DC to use a mode other than "security = user". If a Samba host is configured to use another SMB server or DC in order to validate user connection requests, then it is a fact that some other machine on the network (the "password server") knows more about user than the Samba host. 99% of the time, this other host is a domain controller. Now in order to operate in domain mode security, the "workgroup" parameter must be set to the name of the Windows NT domain (which already has a domain controller, right?)

Therefore configuring a Samba box as a DC for a domain that already by definition has a PDC is asking for trouble. Therefore, you should always configure the Samba DC to be the DMB for its domain.

Chapter 7. Samba Backup Domain Controller to Samba Domain Control

7.1. Prerequisite Reading

Before you continue reading in this chapter, please make sure that you are comfortable with configuring a Samba PDC as described in the Samba-PDC-HOWTO (Samba-PDC-HOWTO.html).

7.2. Background

What is a Domain Controller? It is a machine that is able to answer logon requests from workstations in a Windows NT Domain. Whenever a user logs into a Windows NT Workstation, the workstation connects to a Domain Controller and asks him whether the username and password the user typed in is correct. The Domain Controller replies with a lot of information about the user, for example the place where the users profile is stored, the users full name of the user. All this information is stored in the NT user database, the so-called SAM.

There are two kinds of Domain Controller in a NT 4 compatible Domain: A Primary Domain Controller (PDC) and one or more Backup Domain Controllers (BDC). The PDC contains the master copy of the SAM. Whenever the SAM has to change, for example when a user changes his password, this change has to be done on the PDC. A Backup Domain Controller is a machine that maintains a read-only copy of the SAM. This way it is able to reply to logon requests and authenticate users in case the PDC is not available. During this time no changes to the SAM are possible. Whenever changes to the SAM are done on the PDC, all BDC receive the changes from the PDC.

Since version 2.2 Samba officially supports domain logons for all current Windows Clients, including Windows 2000 and XP. This text assumes the domain to be named SAMBA. To be able to act as a PDC, some parameters in the [global]-section of the smb.conf have to be set:

```
workgroup = SAMBA
domain master = yes
domain logons = yes
```

Several other things like a [homes] and a [netlogon] share also may be set along with settings for the profile path, the users home drive and others. This will not be covered in this document.

7.3. What qualifies a Domain Controller on the network?

Every machine that is a Domain Controller for the domain SAMBA has to register the NetBIOS group name SAMBA#1c with the WINS server and/or by broadcast on the local network. The PDC also registers the unique

NetBIOS name SAMBA#1b with the WINS server. The name type #1b is normally reserved for the domain master browser, a role that has nothing to do with anything related to authentication, but the Microsoft Domain implementation requires the domain master browser to be on the same machine as the PDC.

7.3.1. How does a Workstation find its domain controller?

A NT workstation in the domain SAMBA that wants a local user to be authenticated has to find the domain controller for SAMBA. It does this by doing a NetBIOS name query for the group name SAMBA#1c. It assumes that each of the machines it gets back from the queries is a domain controller and can answer logon requests. To not open security holes both the workstation and the selected (TODO: How is the DC chosen) domain controller authenticate each other. After that the workstation sends the user's credentials (his name and password) to the domain controller, asking for approval.

7.3.2. When is the PDC needed?

Whenever a user wants to change his password, this has to be done on the PDC. To find the PDC, the workstation does a NetBIOS name query for SAMBA#1b, assuming this machine maintains the master copy of the SAM. The workstation contacts the PDC, both mutually authenticate and the password change is done.

7.4. Can Samba be a Backup Domain Controller to an NT PDC?

With version 2.2, no. The native NT SAM replication protocols have not yet been fully implemented. The Samba Team is working on understanding and implementing the protocols, but this work has not been finished for version 2.2.

With version 3.0, the work on both the replication protocols and a suitable storage mechanism has progressed, and some form of NT4 BDC support is expected soon.

Can I get the benefits of a BDC with Samba? Yes. The main reason for implementing a BDC is availability. If the PDC is a Samba machine, a second Samba machine can be set up to service logon requests whenever the PDC is down.

7.5. How do I set up a Samba BDC?

Several things have to be done:

- The domain SID has to be the same on the PDC and the BDC. This used to be stored in the file private/MACHINE.SID. This file is not created anymore since Samba 2.2.5 or even earlier. Nowadays the domain SID is stored in the file private/secrets.tdb. Simply copying the secrets.tdb from the PDC to the BDC does not work, as the BDC would generate a new SID for itself and override the domain SID with this new BDC SID.

To retrieve the domain SID from the PDC or an existing BDC and store it in the secrets.tdb, execute 'net rpc getsid' on the BDC.

- The Unix user database has to be synchronized from the PDC to the BDC. This means that both the /etc/passwd and /etc/group have to be replicated from the PDC to the BDC. This can be done manually whenever changes are made, or the PDC is set up as a NIS master server and the BDC as a NIS slave server. To set up the BDC as a mere NIS client would not be enough, as the BDC would not be able to access its user database in case of a PDC failure.
- The Samba password database in the file private/smbpasswd has to be replicated from the PDC to the BDC. This is a bit tricky, see the next section.
- Any netlogon share has to be replicated from the PDC to the BDC. This can be done manually whenever login scripts are changed, or it can be done automatically together with the smbpasswd synchronization.

Finally, the BDC has to be found by the workstations. This can be done by setting

```
workgroup = samba
domain master = no
domain logons = yes
```

in the [global]-section of the smb.conf of the BDC. This makes the BDC only register the name SAMBA#1c with the WINS server. This is no problem as the name SAMBA#1c is a NetBIOS group name that is meant to be registered by more than one machine. The parameter 'domain master = no' forces the BDC not to register SAMBA#1b which as a unique NetBIOS name is reserved for the Primary Domain Controller.

7.5.1. How do I replicate the smbpasswd file?

Replication of the smbpasswd file is sensitive. It has to be done whenever changes to the SAM are made. Every user's password change is done in the smbpasswd file and has to be replicated to the BDC. So replicating the smbpasswd file very often is necessary.

As the smbpasswd file contains plain text password equivalents, it must not be sent unencrypted over the wire. The best way to set up smbpasswd replication from the PDC to the BDC is to use the utility rsync. rsync can use ssh as a transport. ssh itself can be set up to accept *only* rsync transfer without requiring the user to type a password.

7.5.2. Can I do this all with LDAP?

The simple answer is YES. Samba's pdb_ldap code supports binding to a replica LDAP server, and will also follow referrals and rebind to the master if it ever needs to make a modification to the database. (Normally BDCs are read only, so this will not occur often).

Chapter 8. Samba as a ADS domain member

This is a rough guide to setting up Samba 3.0 with kerberos authentication against a Windows2000 KDC.

8.1. Setup your `smb.conf`

You must use at least the following 3 options in smb.conf:

```
realm = YOUR.KERBEROS.REALM
security = ADS
encrypt passwords = yes
```

In case samba can't figure out your ads server using your realm name, use the **ads server** option in `smb.conf`:

```
ads server = your.kerberos.server
```

> **Note:** You do *not* need a smbpasswd file, and older clients will be authenticated as if **security = domain**, although it won't do any harm and allows you to have local users not in the domain. I expect that the above required options will change soon when we get better active directory integration.

8.2. Setup your `/etc/krb5.conf`

The minimal configuration for `krb5.conf` is:

```
[realms]
    YOUR.KERBEROS.REALM = {
kdc = your.kerberos.server
    }
```

Test your config by doing a **`kinit`** ***`USERNAME@REALM`*** and making sure that your password is accepted by the Win2000 KDC.

> **Note:** The realm must be uppercase.

You also must ensure that you can do a reverse DNS lookup on the IP address of your KDC. Also, the name that this reverse lookup maps to must either be the netbios name of the KDC (ie. the hostname with no domain attached) or it can alternatively be the netbios name followed by the realm.

The easiest way to ensure you get this right is to add a `/etc/hosts` entry mapping the IP address of your KDC to its netbios name. If you don't get this right then you will get a "local error" when you try to join the realm.

If all you want is kerberos support in smbclient then you can skip straight to Test with smbclient now. Creating a computer account and testing your servers is only needed if you want kerberos support for smbd and winbindd.

8.3. Create the computer account

As a user that has write permission on the Samba private directory (usually root) run: **`net ads join`**

8.3.1. Possible errors

"ADS support not compiled in"

> Samba must be reconfigured (remove config.cache) and recompiled (make clean all install) after the kerberos libs and headers are installed.

8.4. Test your server setup

On a Windows 2000 client try **`net use * \\server\share`**. You should be logged in with kerberos without needing to know a password. If this fails then run **`klist tickets`**. Did you get a ticket for the server? Does it have an encoding type of DES-CBC-MD5 ?

8.5. Testing with smbclient

On your Samba server try to login to a Win2000 server or your Samba server using smbclient and kerberos. Use smbclient as usual, but specify the *`-k`* option to choose kerberos authentication.

8.6. Notes

You must change administrator password at least once after DC install, to create the right encoding types w2k doesn’t seem to create the _kerberos._udp and _ldap._tcp in their defaults DNS setup. Maybe fixed in service packs?

Chapter 9. Samba as a NT4 or Win2k domain member

9.1. Joining an NT Domain with Samba 3.0

Assume you have a Samba 3.0 server with a NetBIOS name of `SERV1` and are joining an or Win2k NT domain called `DOM`, which has a PDC with a NetBIOS name of `DOMPDC` and two backup domain controllers with NetBIOS names `DOMBDC1` and `DOMBDC2` .

Firstly, you must edit your `smb.conf` file to tell Samba it should now use domain security.

Change (or add) your *`security =`* (smb.conf.5.html#SECURITY) line in the [global] section of your `smb.conf` to read:

security = domain

Next change the *`workgroup =`* (smb.conf.5.html#WORKGROUP) line in the [global] section to read:

workgroup = DOM

as this is the name of the domain we are joining.

You must also have the parameter *`encrypt passwords`* (smb.conf.5.html#ENCRYPTPASSWORDS) set to `yes` in order for your users to authenticate to the NT PDC.

Finally, add (or modify) a *`password server =`* (smb.conf.5.html#PASSWORDSERVER) line in the [global] section to read:

password server = DOMPDC DOMBDC1 DOMBDC2

These are the primary and backup domain controllers Samba will attempt to contact in order to authenticate users. Samba will try to contact each of these servers in order, so you may want to rearrange this list in order to spread out the authentication load among domain controllers.

Alternatively, if you want smbd to automatically determine the list of Domain controllers to use for authentication, you may set this line to be :

password server = *

This method, allows Samba to use exactly the same mechanism that NT does. This method either broadcasts or uses a WINS database in order to find domain controllers to authenticate against.

In order to actually join the domain, you must run this command:

```
root# net rpc join -S DOMPDC -UAdministrator%password
```

as we are joining the domain DOM and the PDC for that domain (the only machine that has write access to the domain SAM database) is DOMPDC. The `Administrator%password` is the login name and password for an account which has the necessary privilege to add machines to the domain. If this is successful you will see the message:

`Joined domain DOM.` or `Joined 'SERV1' to realm 'MYREALM'`

in your terminal window. See the net(8) (net.8.html) man page for more details.

This process joins the server to thedomain without having to create the machine trust account on the PDC beforehand.

This command goes through the machine account password change protocol, then writes the new (random) machine account password for this Samba server into a file in the same directory in which an smbpasswd file would be stored - normally :

```
/usr/local/samba/private/secrets.tdb
```

This file is created and owned by root and is not readable by any other user. It is the key to the domain-level security for your system, and should be treated as carefully as a shadow password file.

Finally, restart your Samba daemons and get ready for clients to begin using domain security!

9.2. Why is this better than security = server?

Currently, domain security in Samba doesn't free you from having to create local Unix users to represent the users attaching to your server. This means that if domain user `DOM\fred` attaches to your domain security Samba server, there needs to be a local Unix user fred to represent that user in the Unix filesystem. This is very similar to the older Samba security mode security = server (smb.conf.5.html#SECURITYEQUALSSERVER), where Samba would pass through the authentication request to a Windows NT server in the same way as a Windows 95 or Windows 98 server would.

Please refer to the Winbind paper (winbind.html) for information on a system to automatically assign UNIX uids and gids to Windows NT Domain users and groups. This code is available in development branches only at the moment, but will be moved to release branches soon.

The advantage to domain-level security is that the authentication in domain-level security is passed down the authenticated RPC channel in exactly the same way that an NT server would do it. This means Samba servers now participate in domain trust relationships in exactly the same way NT servers do (i.e., you can add Samba servers into a resource domain and have the authentication passed on from a resource domain PDC to an account domain PDC.

In addition, with **security = server** every Samba daemon on a server has to keep a connection open to the authenticating server for as long as that daemon lasts. This can drain the connection resources on a Microsoft NT server and cause it to run out of available connections. With **security = domain**, however, the Samba daemons connect to the PDC/BDC only for as long as is necessary to authenticate the user, and then drop the connection, thus conserving PDC connection resources.

And finally, acting in the same manner as an NT server authenticating to a PDC means that as part of the authentication reply, the Samba server gets the user identification information such as the user SID, the list of NT groups the user belongs to, etc.

> **Note:** Much of the text of this document was first published in the Web magazine LinuxWorld (http://www.linuxworld.com) as the article Doing the NIS/NT Samba (http://www.linuxworld.com/linuxworld/lw-1998-10/lw-10-samba.html).

III. Advanced Configuration

Introduction

Samba has several features that you might want or might not want to use. The chapters in this part each cover one specific feature.

Chapter 10. UNIX Permission Bits and Windows NT Access Control Lists

10.1. Viewing and changing UNIX permissions using the NT security dialogs

Windows NT clients can use their native security settings dialog box to view and modify the underlying UNIX permissions.

Note that this ability is careful not to compromise the security of the UNIX host Samba is running on, and still obeys all the file permission rules that a Samba administrator can set.

> **Note:** All access to Unix/Linux system file via Samba is controlled at the operating system file access control level. When trying to figure out file access problems it is vitally important to identify the identity of the Windows user as it is presented by Samba at the point of file access. This can best be determined from the Samba log files.

10.2. How to view file security on a Samba share

From an NT4/2000/XP client, single-click with the right mouse button on any file or directory in a Samba mounted drive letter or UNC path. When the menu pops-up, click on the *Properties* entry at the bottom of the menu. This brings up the file properties dialog box. Click on the tab *Security* and you will see three buttons, *Permissions*, *Auditing*, and *Ownership*. The *Auditing* button will cause either an error message A requested privilege is not held by the client to appear if the user is not the NT Administrator, or a dialog which is intended to allow an Administrator to add auditing requirements to a file if the user is logged on as the NT Administrator. This dialog is non-functional with a Samba share at this time, as the only useful button, the **Add** button will not currently allow a list of users to be seen.

10.3. Viewing file ownership

Clicking on the **"Ownership"** button brings up a dialog box telling you who owns the given file. The owner name will be of the form :

"SERVER\user (Long name)"

Where *`SERVER`* is the NetBIOS name of the Samba server, *`user`* is the user name of the UNIX user who owns the file, and *`(Long name)`* is the descriptive string identifying the user (normally found in the GECOS field of the UNIX password database). Click on the **Close** button to remove this dialog.

If the parameter *`nt acl support`* is set to `false` then the file owner will be shown as the NT user **"Everyone"**.

The **Take Ownership** button will not allow you to change the ownership of this file to yourself (clicking on it will display a dialog box complaining that the user you are currently logged onto the NT client cannot be found). The reason for this is that changing the ownership of a file is a privileged operation in UNIX, available only to the *root* user. As clicking on this button causes NT to attempt to change the ownership of a file to the current user logged into the NT client this will not work with Samba at this time.

There is an NT chown command that will work with Samba and allow a user with Administrator privilege connected to a Samba server as root to change the ownership of files on both a local NTFS filesystem or remote mounted NTFS or Samba drive. This is available as part of the *Seclib* NT security library written by Jeremy Allison of the Samba Team, available from the main Samba ftp site.

10.4. Viewing file or directory permissions

The third button is the **"Permissions"** button. Clicking on this brings up a dialog box that shows both the permissions and the UNIX owner of the file or directory. The owner is displayed in the form :

"SERVER\user (Long name)"

Where *`SERVER`* is the NetBIOS name of the Samba server, *`user`* is the user name of the UNIX user who owns the file, and *`(Long name)`* is the descriptive string identifying the user (normally found in the GECOS field of the UNIX password database).

If the parameter *`nt acl support`* is set to `false` then the file owner will be shown as the NT user **"Everyone"** and the permissions will be shown as NT "Full Control".

The permissions field is displayed differently for files and directories, so I'll describe the way file permissions are displayed first.

10.4.1. File Permissions

The standard UNIX user/group/world triple and the corresponding "read", "write", "execute" permissions triples are mapped by Samba into a three element NT ACL with the 'r', 'w', and 'x' bits mapped into the corresponding NT permissions. The UNIX world permissions are mapped into the global NT group **Everyone**, followed by the list of permissions allowed for UNIX world. The UNIX owner and group permissions are displayed as an NT **user** icon and an NT **local group** icon respectively followed by the list of permissions allowed for the UNIX user and group.

As many UNIX permission sets don't map into common NT names such as **"read"**, **"change"** or **"full control"** then usually the permissions will be prefixed by the words **"Special Access"** in the NT display list.

But what happens if the file has no permissions allowed for a particular UNIX user group or world component ? In order to allow "no permissions" to be seen and modified then Samba overloads the NT **"Take Ownership"** ACL attribute (which has no meaning in UNIX) and reports a component with no permissions as having the NT **"O"** bit set. This was chosen of course to make it look like a zero, meaning zero permissions. More details on the decision behind this will be given below.

10.4.2. Directory Permissions

Directories on an NT NTFS file system have two different sets of permissions. The first set of permissions is the ACL set on the directory itself, this is usually displayed in the first set of parentheses in the normal **"RW"** NT style. This first set of permissions is created by Samba in exactly the same way as normal file permissions are, described above, and is displayed in the same way.

The second set of directory permissions has no real meaning in the UNIX permissions world and represents the **"inherited"** permissions that any file created within this directory would inherit.

Samba synthesises these inherited permissions for NT by returning as an NT ACL the UNIX permission mode that a new file created by Samba on this share would receive.

10.5. Modifying file or directory permissions

Modifying file and directory permissions is as simple as changing the displayed permissions in the dialog box, and clicking the **OK** button. However, there are limitations that a user needs to be aware of, and also interactions with the standard Samba permission masks and mapping of DOS attributes that need to also be taken into account.

If the parameter *`nt acl support`* is set to `false` then any attempt to set security permissions will fail with an **"Access Denied"** message.

The first thing to note is that the **"Add"** button will not return a list of users in Samba (it will give an error message of **"The remote procedure call failed and did not execute"**). This means that you can only manipulate the current user/group/world permissions listed in the dialog box. This actually works quite well as these are the only permissions that UNIX actually has.

If a permission triple (either user, group, or world) is removed from the list of permissions in the NT dialog box, then when the **"OK"** button is pressed it will be applied as "no permissions" on the UNIX side. If you then view the permissions again the "no permissions" entry will appear as the NT **"O"** flag, as described above. This allows you to add permissions back to a file or directory once you have removed them from a triple component.

As UNIX supports only the "r", "w" and "x" bits of an NT ACL then if other NT security attributes such as "Delete access" are selected then they will be ignored when applied on the Samba server.

When setting permissions on a directory the second set of permissions (in the second set of parentheses) is by default applied to all files within that directory. If this is not what you want you must uncheck the **"Replace permissions on existing files"** checkbox in the NT dialog before clicking **"OK"**.

If you wish to remove all permissions from a user/group/world component then you may either highlight the component and click the **"Remove"** button, or set the component to only have the special **"Take Ownership"** permission (displayed as **"O"**) highlighted.

10.6. Interaction with the standard Samba create mask parameters

There are four parameters to control interaction with the standard Samba create mask parameters. These are :

`security mask`

`force security mode`

`directory security mask`

`force directory security mode`

Once a user clicks **"OK"** to apply the permissions Samba maps the given permissions into a user/group/world r/w/x triple set, and then will check the changed permissions for a file against the bits set in the *`security mask`* (smb.conf.5.html#SECURITYMASK) parameter. Any bits that were changed that are not set to '1' in this parameter are left alone in the file permissions.

Essentially, zero bits in the *`security mask`* mask may be treated as a set of bits the user is *not* allowed to change, and one bits are those the user is allowed to change.

If not set explicitly this parameter is set to the same value as the *`create mask`* (smb.conf.5.html#CREATEMASK) parameter. To allow a user to modify all the user/group/world permissions on a file, set this parameter to 0777.

Next Samba checks the changed permissions for a file against the bits set in the *`force security mode`* (smb.conf.5.html#FORCESECURITYMODE) parameter. Any bits that were changed that correspond to bits set to '1' in this parameter are forced to be set.

Essentially, bits set in the *`force security mode`* parameter may be treated as a set of bits that, when modifying security on a file, the user has always set to be 'on'.

If not set explicitly this parameter is set to the same value as the *`force create mode`* (smb.conf.5.html#FORCECREATEMODE) parameter. To allow a user to modify all the user/group/world permissions on a file with no restrictions set this parameter to 000.

The `security mask` and `force security mode` parameters are applied to the change request in that order.

For a directory Samba will perform the same operations as described above for a file except using the parameter `directory security mask` instead of `security mask`, and `force directory security mode` parameter instead of `force security mode` .

The `directory security mask` parameter by default is set to the same value as the `directory mask` parameter and the `force directory security mode` parameter by default is set to the same value as the `force directory mode` parameter.

In this way Samba enforces the permission restrictions that an administrator can set on a Samba share, whilst still allowing users to modify the permission bits within that restriction.

If you want to set up a share that allows users full control in modifying the permission bits on their files and directories and doesn't force any particular bits to be set 'on', then set the following parameters in the `smb.conf` file in that share specific section :

```
security mask = 0777

force security mode = 0

directory security mask = 0777

force directory security mode = 0
```

10.7. Interaction with the standard Samba file attribute mapping

Samba maps some of the DOS attribute bits (such as "read only") into the UNIX permissions of a file. This means there can be a conflict between the permission bits set via the security dialog and the permission bits set by the file attribute mapping.

One way this can show up is if a file has no UNIX read access for the owner it will show up as "read only" in the standard file attributes tabbed dialog. Unfortunately this dialog is the same one that contains the security info in another tab.

What this can mean is that if the owner changes the permissions to allow themselves read access using the security dialog, clicks **"OK"** to get back to the standard attributes tab dialog, and then clicks **"OK"** on that dialog, then NT will set the file permissions back to read-only (as that is what the attributes still say in the dialog). This means that after setting permissions and clicking **"OK"** to get back to the attributes dialog you should always hit **"Cancel"** rather than **"OK"** to ensure that your changes are not overridden.

Chapter 11. Configuring Group Mapping

Starting with Samba 3.0 alpha 2, a new group mapping function is available. The current method (likely to change) to manage the groups is a new command called smbgroupedit.

The first immediate reason to use the group mapping on a PDC, is that the **domain admin group** of `smb.conf` is now gone. This parameter was used to give the listed users local admin rights on their workstations. It was some magic stuff that simply worked but didn't scale very well for complex setups.

Let me explain how it works on NT/W2K, to have this magic fade away. When installing NT/W2K on a computer, the installer program creates some users and groups. Notably the 'Administrators' group, and gives to that group some privileges like the ability to change the date and time or to kill any process (or close too) running on the local machine. The 'Administrator' user is a member of the 'Administrators' group, and thus 'inherit' the 'Administrators' group privileges. If a 'joe' user is created and become a member of the 'Administrator' group, 'joe' has exactly the same rights as 'Administrator'.

When a NT/W2K machine is joined to a domain, during that phase, the "Domain Administrators' group of the PDC is added to the 'Administrators' group of the workstation. Every members of the 'Domain Administrators' group 'inherit' the rights of the 'Administrators' group when logging on the workstation.

You are now wondering how to make some of your samba PDC users members of the 'Domain Administrators' ? That's really easy.

1. create a unix group (usually in `/etc/group`), let's call it domadm
2. add to this group the users that must be Administrators. For example if you want joe,john and mary, your entry in `/etc/group` will look like:

   ```
   domadm:x:502:joe,john,mary
   ```

3. Map this domadm group to the **domain admins** group by running the command:

   ```
   smbgroupedit -c "Domain Admins" -u domadm
   ```

You're set, joe, john and mary are domain administrators !

Like the Domain Admins group, you can map any arbitrary Unix group to any NT group. You can also make any Unix group a domain group. For example, on a domain member machine (an NT/W2K or a samba server running winbind), you would like to give access to a certain directory to some users who are member of a group on your samba PDC. Flag that group as a domain group by running:

```
smbgroupedit -a unixgroup -td
```

You can list the various groups in the mapping database like this

```
smbgroupedit -v
```

Chapter 12. Printing Support

12.1. Introduction

Beginning with the 2.2.0 release, Samba supports the native Windows NT printing mechanisms implemented via MS-RPC (i.e. the SPOOLSS named pipe). Previous versions of Samba only supported LanMan printing calls.

The additional functionality provided by the new SPOOLSS support includes:

- Support for downloading printer driver files to Windows 95/98/NT/2000 clients upon demand.
- Uploading of printer drivers via the Windows NT Add Printer Wizard (APW) or the Imprints tool set (refer to http://imprints.sourceforge.net).
- Support for the native MS-RPC printing calls such as StartDocPrinter, EnumJobs(), etc... (See the MSDN documentation at http://msdn.microsoft.com/ for more information on the Win32 printing API)
- Support for NT Access Control Lists (ACL) on printer objects
- Improved support for printer queue manipulation through the use of an internal databases for spooled job information

There has been some initial confusion about what all this means and whether or not it is a requirement for printer drivers to be installed on a Samba host in order to support printing from Windows clients. As a side note, Samba does not use these drivers in any way to process spooled files. They are utilized entirely by the clients.

The following MS KB article, may be of some help if you are dealing with Windows 2000 clients: *How to Add Printers with No User Interaction in Windows 2000*

http://support.microsoft.com/support/kb/articles/Q189/1/05.ASP

12.2. Configuration

[print$] vs. [printer$]

Previous versions of Samba recommended using a share named [printer$]. This name was taken from the printer$ service created by Windows 9x clients when a printer was shared. Windows 9x printer servers always have a printer$ service which provides read-only access via no password in order to support printer driver downloads.

However, the initial implementation allowed for a parameter named *`printer driver location`* to be used on a per share basis to specify the location of the driver files associated with that printer. Another parameter named *`printer driver`* provided a means of defining the printer driver name to be sent to the client.

12.2.1. Creating [print$]

In order to support the uploading of printer driver files, you must first configure a file share named [print$]. The name of this share is hard coded in Samba's internals so the name is very important (print$ is the service used by Windows NT print servers to provide support for printer driver download).

You should modify the server's smb.conf file to add the global parameters and to create the following file share (of course, some of the parameter values, such as 'path' are arbitrary and should be replaced with appropriate values for your site):

```
[global]
    ; members of the ntadmin group should be able
    ; to add drivers and set printer properties
    ; root is implicitly a 'printer admin'
    printer admin = @ntadmin

[print$]
    path = /usr/local/samba/printers
    guest ok = yes
    browseable = yes
    read only = yes
    ; since this share is configured as read only, then we need
    ; a 'write list'.  Check the file system permissions to make
    ; sure this account can copy files to the share.  If this
    ; is setup to a non-root account, then it should also exist
    ; as a 'printer admin'
    write list = @ntadmin,root
```

The `write list` (smb.conf.5.html#WRITELIST) is used to allow administrative level user accounts to have write access in order to update files on the share. See the smb.conf(5) man page (smb.conf.5.html) for more information on configuring file shares.

The requirement for **guest ok = yes** (smb.conf.5.html#GUESTOK) depends upon how your site is configured. If users will be guaranteed to have an account on the Samba host, then this is a non-issue.

> **Author's Note:** The non-issue is that if all your Windows NT users are guaranteed to be authenticated by the Samba server (such as a domain member server and the NT user has already been validated by the Domain Controller in order to logon to the Windows NT console), then guest access is not necessary. Of course, in a workgroup environment where you just want to be able to print without worrying about silly accounts and security, then configure the share for guest access. You'll probably want to add **map to guest = Bad User** (smb.conf.5.html#MAPTOGUEST) in the [global] section as well. Make sure you understand what this parameter does before using it though. --jerry

In order for a Windows NT print server to support the downloading of driver files by multiple client architectures, it must create subdirectories within the [print$] service which correspond to each of the supported client architectures. Samba follows this model as well.

Next create the directory tree below the [print$] share for each architecture you wish to support.

```
[print$]----- |-W32X86 ; "Windows NT x86" |-WIN40 ; "Windows 95/98" |-W32ALPHA ;
"Windows NT Alpha_AXP" |-W32MIPS ; "Windows NT R4000" |-W32PPC ; "Windows NT
PowerPC"
```

ATTENTION! REQUIRED PERMISSIONS

In order to currently add a new driver to you Samba host, one of two conditions must hold true:

- The account used to connect to the Samba host must have a uid of 0 (i.e. a root account)
- The account used to connect to the Samba host must be a member of the *`printer admin`* (smb.conf.5.html#PRINTERADMIN) list.

Of course, the connected account must still possess access to add files to the subdirectories beneath [print$]. Remember that all file shares are set to ’read only’ by default.

Once you have created the required [print$] service and associated subdirectories, simply log onto the Samba server using a root (or *`printer admin`*) account from a Windows NT 4.0/2k client. Open "Network Neighbourhood" or "My Network Places" and browse for the Samba host. Once you have located the server, navigate to the "Printers..." folder. You should see an initial listing of printers that matches the printer shares defined on your Samba host.

12.2.2. Setting Drivers for Existing Printers

The initial listing of printers in the Samba host’s Printers folder will have no real printer driver assigned to them. This defaults to a NULL string to allow the use of the local Add Printer Wizard on NT/2000 clients. Attempting to view the printer properties for a printer which has this default driver assigned will result in the error message:

Device settings cannot be displayed. The driver for the specified printer is not installed, only spooler properties will be displayed. Do you want to install the driver now?

Click "No" in the error dialog and you will be presented with the printer properties window. The way to assign a driver to a printer is to either

- Use the "New Driver..." button to install a new printer driver, or
- Select a driver from the popup list of installed drivers. Initially this list will be empty.

If you wish to install printer drivers for client operating systems other than "Windows NT x86", you will need to use the "Sharing" tab of the printer properties dialog.

Assuming you have connected with a root account, you will also be able modify other printer properties such as ACLs and device settings using this dialog box.

A few closing comments for this section, it is possible on a Windows NT print server to have printers listed in the Printers folder which are not shared. Samba does not make this distinction. By definition, the only printers of which Samba is aware are those which are specified as shares in `smb.conf`.

Another interesting side note is that Windows NT clients do not use the SMB printer share, but rather can print directly to any printer on another Windows NT host using MS-RPC. This of course assumes that the printing client has the necessary privileges on the remote host serving the printer. The default permissions assigned by Windows NT to a printer gives the "Print" permissions to the "Everyone" well-known group.

12.2.3. Support a large number of printers

One issue that has arisen during the development phase of Samba 2.2 is the need to support driver downloads for 100's of printers. Using the Windows NT APW is somewhat awkward to say the list. If more than one printer are using the same driver, the **rpcclient's setdriver command** (rpcclient.1.html) can be used to set the driver associated with an installed driver. The following is example of how this could be accomplished:

```
$ rpcclient pogo -U root%secret -c "enumdrivers"

Domain=[NARNIA] OS=[Unix] Server=[Samba 2.2.0-alpha3]

[Windows NT x86]
Printer Driver Info 1:
     Driver Name: [HP LaserJet 4000 Series PS]

Printer Driver Info 1:
     Driver Name: [HP LaserJet 2100 Series PS]

Printer Driver Info 1:
     Driver Name: [HP LaserJet 4Si/4SiMX PS]

$ rpcclient pogo -U root%secret -c "enumprinters"

Domain=[NARNIA] OS=[Unix] Server=[Samba 2.2.0-alpha3]
     flags:[0x800000]
     name:[\\POGO\hp-print]
     description:[POGO\\POGO\hp-print,NO DRIVER AVAILABLE FOR THIS PRINTER,]
     comment:[]


$ rpcclient pogo -U root%secret -c "setdriver hp-print \"HP LaserJet 4000
Series PS\""

Domain=[NARNIA] OS=[Unix] Server=[Samba 2.2.0-alpha3]
Successfully set hp-print to driver HP LaserJet 4000 Series PS.
```

12.2.4. Adding New Printers via the Windows NT APW

By default, Samba offers all printer shares defined in `smb.conf` in the "Printers..." folder. Also existing in this folder is the Windows NT Add Printer Wizard icon. The APW will be show only if

- The connected user is able to successfully execute an OpenPrinterEx(\\server) with administrative privileges (i.e. root or *`printer admin`*).
- *`show add printer wizard = yes`* (smb.conf.5.html#SHOWADDPRINTERWIZARD) (the default).

In order to be able to use the APW to successfully add a printer to a Samba server, the *`add printer command`* (smb.conf.5.html#ADDPRINTERCOMMAND) must have a defined value. The program hook must successfully add the printer to the system (i.e. `/etc/printcap` or appropriate files) and `smb.conf` if necessary.

When using the APW from a client, if the named printer share does not exist, **smbd** will execute the *`add printer command`* and reparse to the `smb.conf` to attempt to locate the new printer share. If the share is still not defined, an error of "Access Denied" is returned to the client. Note that the *`add printer program`* is executed under the context of the connected user, not necessarily a root account.

There is a complementary *`delete printer command`* (smb.conf.5.html#DELETEPRINTERCOMMAND) for removing entries from the "Printers..." folder.

The following is an example *`add printer command`* (smb.conf.5.html#ADDPRINTERCOMMAN) script. It adds the appropriate entries to `/etc/printcap.local` (change that to what you need) and returns a line of ’Done’ which is needed for the whole process to work.

```
#!/bin/sh

# Script to insert a new printer entry into printcap.local
#
# $1, printer name, used as the descriptive name
# $2, share name, used as the printer name for Linux
# $3, port name
# $4, driver name
# $5, location, used for the device file of the printer
# $6, win9x location

#
# Make sure we use the location that RedHat uses for local printer defs
PRINTCAP=/etc/printcap.local
DATE=`date +%Y%m%d-%H%M%S`
LP=lp
RESTART="service lpd restart"

# Keep a copy
cp $PRINTCAP $PRINTCAP.$DATE
# Add the printer to $PRINTCAP
echo ""   >> $PRINTCAP
echo "$2|$1:\\"  >> $PRINTCAP
echo "  :sd=/var/spool/lpd/$2:\\"  >> $PRINTCAP
echo "  :mx=0:ml=0:sh:\\"  >> $PRINTCAP
echo "  :lp=/usr/local/samba/var/print/$5.prn:"  >> $PRINTCAP
```

```
touch "/usr/local/samba/var/print/$5.prn" >> /tmp/printadd.$$ 2>&1
chown $LP "/usr/local/samba/var/print/$5.prn" >> /tmp/printadd.$$ 2>&1

mkdir /var/spool/lpd/$2
chmod 700 /var/spool/lpd/$2
chown $LP /var/spool/lpd/$2
#echo $1 >> "/usr/local/samba/var/print/$5.prn"
#echo $2 >> "/usr/local/samba/var/print/$5.prn"
#echo $3 >> "/usr/local/samba/var/print/$5.prn"
#echo $4 >> "/usr/local/samba/var/print/$5.prn"
#echo $5 >> "/usr/local/samba/var/print/$5.prn"
#echo $6 >> "/usr/local/samba/var/print/$5.prn"
$RESTART >> "/usr/local/samba/var/print/$5.prn"
# Not sure if this is needed
touch /usr/local/samba/lib/smb.conf
#
# You need to return a value, but I am not sure what it means.
#
echo "Done"
exit 0
```

12.2.5. Samba and Printer Ports

Windows NT/2000 print servers associate a port with each printer. These normally take the form of LPT1:, COM1:, FILE:, etc... Samba must also support the concept of ports associated with a printer. By default, only one printer port, named "Samba Printer Port", exists on a system. Samba does not really a port in order to print, rather it is a requirement of Windows clients.

Note that Samba does not support the concept of "Printer Pooling" internally either. This is when a logical printer is assigned to multiple ports as a form of load balancing or fail over.

If you require that multiple ports be defined for some reason, `smb.conf` possesses a *`enumports command`* (smb.conf.5.html#ENUMPORTSCOMMAND) which can be used to define an external program that generates a listing of ports on a system.

12.3. The Imprints Toolset

The Imprints tool set provides a UNIX equivalent of the Windows NT Add Printer Wizard. For complete information, please refer to the Imprints web site at http://imprints.sourceforge.net/ (http://imprints.sourceforge.net/) as well as the documentation included with the imprints source distribution. This section will only provide a brief introduction to the features of Imprints.

12.3.1. What is Imprints?

Imprints is a collection of tools for supporting the goals of

- Providing a central repository information regarding Windows NT and 95/98 printer driver packages
- Providing the tools necessary for creating the Imprints printer driver packages.
- Providing an installation client which will obtain and install printer drivers on remote Samba and Windows NT 4 print servers.

12.3.2. Creating Printer Driver Packages

The process of creating printer driver packages is beyond the scope of this document (refer to Imprints.txt also included with the Samba distribution for more information). In short, an Imprints driver package is a gzipped tarball containing the driver files, related INF files, and a control file needed by the installation client.

12.3.3. The Imprints server

The Imprints server is really a database server that may be queried via standard HTTP mechanisms. Each printer entry in the database has an associated URL for the actual downloading of the package. Each package is digitally signed via GnuPG which can be used to verify that package downloaded is actually the one referred in the Imprints database. It is *not* recommended that this security check be disabled.

12.3.4. The Installation Client

More information regarding the Imprints installation client is available in the `Imprints-Client-HOWTO.ps` file included with the imprints source package.

The Imprints installation client comes in two forms.

- a set of command line Perl scripts
- a GTK+ based graphical interface to the command line perl scripts

The installation client (in both forms) provides a means of querying the Imprints database server for a matching list of known printer model names as well as a means to download and install the drivers on remote Samba and Windows NT print servers.

The basic installation process is in four steps and perl code is wrapped around **smbclient** and **rpcclient**.

```
foreach (supported architecture for a given driver)
{
     1.  rpcclient: Get the appropriate upload directory
         on the remote server
     2.  smbclient: Upload the driver files
     3.  rpcclient: Issues an AddPrinterDriver() MS-RPC
}

4.  rpcclient: Issue an AddPrinterEx() MS-RPC to actually
    create the printer
```

One of the problems encountered when implementing the Imprints tool set was the name space issues between various supported client architectures. For example, Windows NT includes a driver named "Apple LaserWriter II NTX v51.8" and Windows 95 calls its version of this driver "Apple LaserWriter II NTX"

The problem is how to know what client drivers have been uploaded for a printer. As astute reader will remember that the Windows NT Printer Properties dialog only includes space for one printer driver name. A quick look in the Windows NT 4.0 system registry at

```
HKLM\System\CurrentControlSet\Control\Print\Environment
```

will reveal that Windows NT always uses the NT driver name. This is ok as Windows NT always requires that at least the Windows NT version of the printer driver is present. However, Samba does not have the requirement internally. Therefore, how can you use the NT driver name if is has not already been installed?

The way of sidestepping this limitation is to require that all Imprints printer driver packages include both the Intel Windows NT and 95/98 printer drivers and that NT driver is installed first.

12.4. Diagnosis

12.4.1. Introduction

This is a short description of how to debug printing problems with Samba. This describes how to debug problems with printing from a SMB client to a Samba server, not the other way around. For the reverse see the examples/printing directory.

Ok, so you want to print to a Samba server from your PC. The first thing you need to understand is that Samba does not actually do any printing itself, it just acts as a middleman between your PC client and your Unix printing subsystem. Samba receives the file from the PC then passes the file to a external "print command". What print command you use is up to you.

The whole things is controlled using options in smb.conf. The most relevant options (which you should look up in the smb.conf man page) are:

```
[global]
  print command     - send a file to a spooler
  lpq command       - get spool queue status
  lprm command      - remove a job
[printers]
  path = /var/spool/lpd/samba
```

The following are nice to know about:

```
        queuepause command   - stop a printer or print queue
        queueresume command  - start a printer or print queue
```

Example:

```
        print command = /usr/bin/lpr -r -P%p %s
        lpq command   = /usr/bin/lpq    -P%p %s
        lprm command  = /usr/bin/lprm   -P%p %j
        queuepause command = /usr/sbin/lpc -P%p stop
        queuepause command = /usr/sbin/lpc -P%p start
```

Samba should set reasonable defaults for these depending on your system type, but it isn't clairvoyant. It is not uncommon that you have to tweak these for local conditions. The commands should always have fully specified pathnames, as the smdb may not have the correct PATH values.

When you send a job to Samba to be printed, it will make a temporary copy of it in the directory specified in the [printers] section. and it should be periodically cleaned out. The lpr -r option requests that the temporary copy be removed after printing; If printing fails then you might find leftover files in this directory, and it should be periodically cleaned out. Samba used the lpq command to determine the "job number" assigned to your print job by the spooler.

The %>letter< are "macros" that get dynamically replaced with appropriate values when they are used. The %s gets replaced with the name of the spool file that Samba creates and the %p gets replaced with the name of the printer. The %j gets replaced with the "job number" which comes from the lpq output.

12.4.2. Debugging printer problems

One way to debug printing problems is to start by replacing these command with shell scripts that record the arguments and the contents of the print file. A simple example of this kind of things might be:

```
print command = /tmp/saveprint %p %s

    #!/bin/saveprint
    # we make sure that we are the right user
    /usr/bin/id -p >/tmp/tmp.print
    # we run the command and save the error messages
    # replace the command with the one appropriate for your system
    /usr/bin/lpr -r -P$1 $2 2>>&/tmp/tmp.print
```

Then you print a file and try removing it. You may find that the print queue needs to be stopped in order to see the queue status and remove the job:

```
h4: {42} % echo hi >/tmp/hi
h4: {43} % smbclient //localhost/lw4
added interface ip=10.0.0.4 bcast=10.0.0.255 nmask=255.255.255.0
Password:
Domain=[ASTART] OS=[Unix] Server=[Samba 2.0.7]
smb: \> print /tmp/hi
putting file /tmp/hi as hi-17534 (0.0 kb/s) (average 0.0 kb/s)
smb: \> queue
1049     3           hi-17534
smb: \> cancel 1049
Error cancelling job 1049 : code 0
smb: \> cancel 1049
Job 1049 cancelled
smb: \> queue
smb: \> exit
```

The 'code 0' indicates that the job was removed. The comment by the smbclient is a bit misleading on this. You can observe the command output and then and look at the /tmp/tmp.print file to see what the results are. You can quickly find out if the problem is with your printing system. Often people have problems with their /etc/printcap file or permissions on various print queues.

12.4.3. What printers do I have?

You can use the 'testprns' program to check to see if the printer name you are using is recognized by Samba. For example, you can use:

```
    testprns printer /etc/printcap
```

Samba can get its printcap information from a file or from a program. You can try the following to see the format of the extracted information:

```
    testprns -a printer /etc/printcap

    testprns -a printer '|/bin/cat printcap'
```

12.4.4. Setting up printcap and print servers

You may need to set up some printcaps for your Samba system to use. It is strongly recommended that you use the facilities provided by the print spooler to set up queues and printcap information.

Samba requires either a printcap or program to deliver printcap information. This printcap information has the format:

```
name|alias1|alias2...:option=value:...
```

For almost all printing systems, the printer 'name' must be composed only of alphanumeric or underscore '_' characters. Some systems also allow hyphens ('-') as well. An alias is an alternative name for the printer, and an alias with a space in it is used as a 'comment' about the printer. The printcap format optionally uses a \ at the end of lines to extend the printcap to multiple lines.

Here are some examples of printcap files:

1. pr just printer name
2. pr|alias printer name and alias
3. pr|My Printer printer name, alias used as comment
4. pr:sh:\ Same as pr:sh:cm= testing :cm= \ testing
5. pr:sh Same as pr:sh:cm= testing :cm= testing

Samba reads the printcap information when first started. If you make changes in the printcap information, then you must do the following:

1. make sure that the print spooler is aware of these changes. The LPRng system uses the 'lpc reread' command to do this.
2. make sure that the spool queues, etc., exist and have the correct permissions. The LPRng system uses the 'checkpc -f' command to do this.
3. You now should send a SIGHUP signal to the smbd server to have it reread the printcap information.

12.4.5. Job sent, no output

This is the most frustrating part of printing. You may have sent the job, verified that the job was forwarded, set up a wrapper around the command to send the file, but there was no output from the printer.

First, check to make sure that the job REALLY is getting to the right print queue. If you are using a BSD or LPRng print spooler, you can temporarily stop the printing of jobs. Jobs can still be submitted, but they will not be printed. Use:

```
lpc -Pprinter stop
```

Now submit a print job and then use ’lpq -Pprinter’ to see if the job is in the print queue. If it is not in the print queue then you will have to find out why it is not being accepted for printing.

Next, you may want to check to see what the format of the job really was. With the assistance of the system administrator you can view the submitted jobs files. You may be surprised to find that these are not in what you would expect to call a printable format. You can use the UNIX ’file’ utitily to determine what the job format actually is:

```
    cd /var/spool/lpd/printer   # spool directory of print jobs
    ls                          # find job files
    file dfA001myhost
```

You should make sure that your printer supports this format OR that your system administrator has installed a ’print filter’ that will convert the file to a format appropriate for your printer.

12.4.6. Job sent, strange output

Once you have the job printing, you can then start worrying about making it print nicely.

The most common problem is extra pages of output: banner pages OR blank pages at the end.

If you are getting banner pages, check and make sure that the printcap option or printer option is configured for no banners. If you have a printcap, this is the :sh (suppress header or banner page) option. You should have the following in your printer.

```
printer: ... :sh
```

If you have this option and are still getting banner pages, there is a strong chance that your printer is generating them for you automatically. You should make sure that banner printing is disabled for the printer. This usually requires using the printer setup software or procedures supplied by the printer manufacturer.

If you get an extra page of output, this could be due to problems with your job format, or if you are generating PostScript jobs, incorrect setting on your printer driver on the MicroSoft client. For example, under Win95 there is a option:

```
Printers|Printer Name|(Right Click)Properties|Postscript|Advanced|
```

that allows you to choose if a Ctrl-D is appended to all jobs. This is a very bad thing to do, as most spooling systems will automatically add a ^D to the end of the job if it is detected as PostScript. The multiple ^D may cause an additional page of output.

12.4.7. Raw PostScript printed

This is a problem that is usually caused by either the print spooling system putting information at the start of the print job that makes the printer think the job is a text file, or your printer simply does not support PostScript. You may need to enable ’Automatic Format Detection’ on your printer.

12.4.8. Advanced Printing

Note that you can do some pretty magic things by using your imagination with the "print command" option and some shell scripts. Doing print accounting is easy by passing the %U option to a print command shell script. You could even make the print command detect the type of output and its size and send it to an appropriate printer.

12.4.9. Real debugging

If the above debug tips don’t help, then maybe you need to bring in the bug guns, system tracing. See Tracing.txt in this directory.

Chapter 13. CUPS Printing Support

13.1. Introduction

The Common Unix Print System (CUPS) has become very popular, but to many it is a very mystical tool. There is a great deal of uncertainty regarding CUPS and how it works. The result is seen in a large number of posting on the samba mailing lists expressing frustration when MS Windows printers appear not to work with a CUPS backr-end. /para>

This is a good time to point out how CUPS can be used and what it does. CUPS is more than just a print spooling system - it is a complete printer management system that complies with HTTP and IPP protocols. It can be managed remotely via a web browser and it can print using http and ipp protocols.

CUPS allows to creation of RAW printers (ie: NO file format translation) as well as SMART printers (ie: CUPS does file format conversion as required for the printer). In many ways this gives CUPS similar capabilities to the MS Windows print monitoring system. Of course, if you are a CUPS advocate, you would agrue that CUPS is better! In any case, let us now move on to explore how one may configure CUPS for interfacing with MS Windows print clients via Samba.

CUPS (http://www.cups.org/) is a newcomer in the UNIX printing scene, which has convinced many people upon first trial already. However, it has quite a few new features, which make it different from other, more traditional printing systems.

13.2. Configuring `smb.conf` for CUPS

Printing with CUPS in the most basic `smb.conf` setup in Samba-3 only needs two settings: **printing = cups** and **printcap = cups**. While CUPS itself doesn't need a printcap anymore, the `cupsd.conf` configuration file knows two directives (example: **Printcap /etc/printcap** and **PrintcapFormat BSD**), which control if such a file should be created for the convenience of third party applications. Make sure it is set! For details see **man cupsd.conf** and other CUPS-related documentation.

If SAMBA is compiled against libcups, then **printcap = cups** uses the CUPS API to list printers, submit jobs, etc. Otherwise it maps to the System V commands with an additional *-oraw* option for printing. On a Linux system, you can use the **ldd** command to find out details (ldd may not be present on other OS platforms, or its function may be embodied by a different command):

```
transmeta:/home/kurt # ldd `which smbd`
        libssl.so.0.9.6 => /usr/lib/libssl.so.0.9.6 (0x4002d000)
        libcrypto.so.0.9.6 => /usr/lib/libcrypto.so.0.9.6 (0x4005a000)
        libcups.so.2 => /usr/lib/libcups.so.2 (0x40123000)
        libdl.so.2 => /lib/libdl.so.2 (0x401e8000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x401ec000)
        libpam.so.0 => /lib/libpam.so.0 (0x40202000)
        libc.so.6 => /lib/libc.so.6 (0x4020b000)
```

```
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

The line "libcups.so.2 => /usr/lib/libcups.so.2 (0x40123000)" shows there is CUPS support compiled into this version of Samba. If this is the case, and **printing = cups** is set, then any otherwise manually set print command in `smb.conf` is ignored.

13.3. CUPS - RAW Print Through Mode

> **Note:** When used in raw print through mode is will be necessary to use the printer vendor's drivers in each Windows client PC.

When CUPS printers are configured for RAW print-through mode operation it is the responsibility of the Samba client to fully render the print job (file) in a format that is suitable for direct delivery to the printer. In this case CUPS will NOT do any print file format conversion work.

The CUPS files that need to be correctly set for RAW mode printers to work are:

- `/etc/cups/mime.types`
- `/etc/cups/mime.convs`

Both contain entries that must be uncommented to allow *RAW* mode operation.

Firstly, to enable CUPS based printing from Samba the following options must be enabled in your `smb.conf` file [globals] section:

- printing = CUPS
- printcap = CUPS

When these parameters are specified the print directives in `smb.conf` (as well as in samba itself) will be ignored because samba will directly interface with CUPS through it's application program interface (API) - so long as Samba has been compiled with CUPS library (libcups) support. If samba has NOT been compiled with CUPS support then printing will use the System V AT&T command set with the *-oraw* option automatically passing through.

Cupsomatic (an enhanced printing utility that is part of some CUPS implementations) on the Samba/CUPS server does *not* add any features if a file is really printed "raw". However, if you have loaded the driver for the Windows client from the CUPS server, using the "cupsaddsmb" utility, and if this driver is one using a "Foomatic" PPD, the PJL header in question is already added on the Windows client, at the time when the driver initially generated the PostScript data and CUPS in true "-oraw" manner doesn't remove this PJL header and passes the file "as is" to its printer communication backend.

> **Note:** NOTE: editing in the "mime.convs" and the "mime.types" file does not *enforce* "raw" printing, it only *allows* it.

Print files that arrive from MS Windows printing are "auto-typed" by CUPS. This aids the process of determining proper treatment while in the print queue system.

- Files generated by PCL drivers and directed at PCK printers get auto-typed as `application/octet-stream`. Unknown file format types also get auto-typed with this tag.
- Files generated by a Postscript driver and directed at a Postscript printer are auto-typed depending on the auto-detected most suitable MIME type as:
  - * application/postscript
  - * application/vnd.cups-postscript

"application/postscript" first goes thru the "pstops" filter (where the page counting and accounting takes place). The outcome will be of MIME type "application/vnd.cups-postscript". The pstopsfilter reads and uses information from the PPD and inserts user-provided options into the PostScript file. As a consequence, the filtered file could possibly have an unwanted PJL header.

"application/postscript" will be all files with a ".ps", ".ai", ".eps" suffix or which have as their first character string one of "%!" or "<04>%".

"application/vnd.cups-postscript" will files which contain the string "LANGUAGE=POSTSCRIPT" (or similar variations with different capitalization) in the first 512 bytes, and also contain the "PJL super escape code" in the first 128 bytes ("<1B>%-12345X"). Very likely, most PostScript files generated on Windows using a CUPS or other PPD, will have to be auto-typed as "vnd.cups-postscript". A file produced with a "Generic PostScript driver" will just be tagged "application/postscript".

Once the file is in "application/vnd.cups-postscript" format, either "pstoraster" or "cupsomatic" will take over (depending on the printer configuration, as determined by the PPD in use).

> **Note:** A printer queue with *no* PPD associated to it is a "raw" printer and all files will go directly there as received by the spooler. The exeptions are file types "application/octet-stream" which need "passthrough feature" enabled. "Raw" queues don't do any filtering at all, they hand the file directly to the CUPS backend. This backend is responsible for the sending of the data to the device (as in the "device URI" notation as lpd://, socket://, smb://, ipp://, http://, parallel:/, serial:/, usb:/ etc.)

> **Note:** "cupsomatic"/Foomatic are *not* native CUPS drivers and they don't ship with CUPS. They are a Third Party add-on, developed at Linuxprinting.org. As such, they are a brilliant hack to make all models (driven by Ghostscript drivers/filters in traditional spoolers) also work via CUPS, with the same (good or bad!) quality as in these other spoolers. "cupsomatic" is only a vehicle to execute a ghostscript commandline at that stage in the CUPS filtering chain, where "normally" the native CUPS "pstoraster" filter would kick in. cupsomatic by-passes pstoraster, "kidnaps" the printfile from CUPS away and re-directs it to go through Ghostscipt. CUPS accepts this, because the associated CUPS-O-Matic-/Foomatic-PPD specifies:

```
*cupsFilter:  "application/vnd.cups-postscript 0 cupsomatic"
```

> This line persuades CUPS to hand the file to cupsomatic, once it has successfully converted it to the MIME type "application/vnd.cups-postscript". This conversion will not happen for Jobs arriving from Windows which are auto-typed "application/octet-stream", with the according changes in "/etc/cups/mime.types" in place.

CUPS is widely configurable and flexible, even regarding its filtering mechanism. Another workaround in some situations would be to have in "/etc/cups/mime.types" entries as follows:

```
application/postscript           application/vnd.cups-raw  0  -
application/vnd.cups-postscript  application/vnd.cups-raw  0  -
```

This would prevent all Postscript files from being filtered (rather, they will go thru the virtual "nullfilter" denoted with "-"). This could only be useful for PS printers. If you want to print PS code on non-PS printers an entry as follows could be useful:

```
*/*           application/vnd.cups-raw  0  -
```

and would effectively send *all* files to the backend without further processing.

Lastly, you could have the following entry:

```
application/vnd.cups-postscript  application/vnd.cups-raw  0  my_PJL_stripping_filter
```

You will need to write a "my_PJL_stripping_filter" (could be a shellscript) that parses the PostScript and removes the unwanted PJL. This would need to conform to CUPS filter design (mainly, receive and pass the parameters printername, job-id, username, jobtitle, copies, print options and possibly the filename). It would be installed as world executable into "/usr/lib/cups/filters/" and will be called by CUPS if it encounters a MIME type "application/vnd.cups-postscript".

CUPS can handle "-o job-hold-until=indefinite". This keeps the job in the queue "on hold". It will only be printed upon manual release by the printer operator. This is a requirement in many "central reproduction departments", where a few operators manage the jobs of hundreds of users on some big machine, where no user is allowed to have direct access. (The operators often need to load the proper paper type before running the 10.000 page job requested by marketing for the mailing, etc.).

13.4. CUPS as a network PostScript RIP -- CUPS drivers working on server, Adobe PostScript driver with CUPS-PPDs downloaded to clients

CUPS is perfectly able to use PPD files (PostScript Printer Descriptions). PPDs can control all print device options. They are usually provided by the manufacturer -- if you own a PostSript printer, that is. PPD files are always a component of PostScript printer drivers on MS Windows or Apple Mac OS systems. They are ASCII files containing user-selectable print options, mapped to appropriate PostScript, PCL or PJL commands for the target printer. Printer driver GUI dialogs translate these options "on-the-fly" into buttons and drop-down lists for the user to select.

CUPS can load, without any conversions, the PPD file from any Windows (NT is recommended) PostScript driver and handle the options. There is a web browser interface to the print options (select http://localhost:631/printers/ and click on one "Configure Printer" button to see it), a commandline interface (see **man lpoptions** or try if you have **lphelp** on your system) plus some different GUI frontends on Linux UNIX, which can present PPD options to the users. PPD options are normally meant to become evaluated by the PostScript RIP on the real PostScript printer.

CUPS doesn't stop at "real" PostScript printers in its usage of PPDs. The CUPS developers have extended the PPD concept, to also describe available device and driver options for non-PostScript printers through CUPS-PPDs.

This is logical, as CUPS includes a fully featured PostScript interpreter (RIP). This RIP is based on Ghostscript. It can process all received PostScript (and additionally many other file formats) from clients. All CUPS-PPDs geared to non-PostScript printers contain an additional line, starting with the keyword *`*cupsFilter`*. This line tells the CUPS print system which printer-specific filter to use for the interpretation of the accompanying PostScript. Thus CUPS lets all its printers appear as PostScript devices to its clients, because it can act as a PostScript RIP for those printers, processing the received PostScript code into a proper raster print format.

CUPS-PPDs can also be used on Windows-Clients, on top of a PostScript driver (recommended is the Adobe one).

This feature enables CUPS to do a few tricks no other spooler can do:

- act as a networked PostScript RIP (Raster Image Processor), handling printfiles from all client platforms in a uniform way;
- act as a central accounting and billing server, as all files are passed through the **pstops** Filter and are therefor logged in the CUPS `page_log`. - *NOTE:* this can not happen with "raw" print jobs, which always remain unfiltered per definition;
- enable clients to consolidate on a single PostScript driver, even for many different target printers.

13.5. Windows Terminal Servers (WTS) as CUPS clients

This setup may be of special interest to people experiencing major problems in WTS environments. WTS need often a multitude of non-PostScript drivers installed to run their clients' variety of different printer models. This often imposes the price of much increased instability. In many cases, in an attempt to overcome this problem, site administrators have resorted to restrict the allowed drivers installed on their WTS to one generic PCL- and one PostScript driver. This however restricts the clients in the amount of printer options available for them -- often they can't get out more then simplex prints from one standard paper tray, while their devices could do much better, if driven by a different driver!

Using an Adobe PostScript driver, enabled with a CUPS-PPD, seems to be a very elegant way to overcome all these shortcomings. The PostScript driver is not known to cause major stability problems on WTS (even if used with many different PPDs). The clients will be able to (again) chose paper trays, duplex printing and other settings. However, there is a certain price for this too: a CUPS server acting as a PostScript RIP for its clients requires more CPU and RAM than just to act as a "raw spooling" device. Plus, this setup is not yet widely tested, although the first feedbacks look very promising...

13.6. Setting up CUPS for driver download

The **cupsadsmb** utility (shipped with all current CUPS versions) makes the sharing of any (or all) installed CUPS printers very easy. Prior to using it, you need the following settings in `smb.conf`:

```
[global]
 load printers = yes
 printing = cups
 printcap name = cups

[printers]
 comment = All Printers
 path = /var/spool/samba
 browseable = no
 public = yes
 guest ok = yes
 writable = no
 printable = yes
 printer admin = root

[print$]
 comment = Printer Drivers
 path = /etc/samba/drivers
 browseable = yes
 guest ok = no
 read only = yes
 write list = root
```

For licensing reasons the necessary files of the Adobe Postscript driver can not be distributed with either Samba or CUPS. You need to download them yourself from the Adobe website. Once extracted, create a `drivers` directory in the CUPS data directory (usually `/usr/share/cups/`). Copy the Adobe files using UPPERCASE filenames, to this directory as follows:

```
ADFONTS.MFM
ADOBEPS4.DRV
ADOBEPS4.HLP
ADOBEPS5.DLL
ADOBEPSU.DLL
ADOBEPSU.HLP
DEFPRTR2.PPD
ICONLIB.DLL
```

Users of the ESP Print Pro software are able to install their "Samba Drivers" package for this purpose with no problem.

13.7. Sources of CUPS drivers / PPDs

On the internet you can find now many thousand CUPS-PPD files (with their companion filters), in many national languages, supporting more than 1.000 non-PostScript models.

- ESP PrintPro (http://wwwl.easysw.com/printpro/) (http://wwwl.easysw.com/printpro/) (commercial, non-Free) is packaged with more than 3.000 PPDs, ready for successful usage "out of the box" on Linux, IBM-AIX, HP-UX, Sun-Solaris, SGI-IRIX, Compaq Tru64, Digital Unix and some more commercial Unices (it is written by the CUPS developers themselves and its sales help finance the further development of CUPS, as they feed their creators)
- the Gimp-Print-Project (http://gimp-print.sourceforge.net/) (http://gimp-print.sourceforge.net/) (GPL, Free Software) provides around 120 PPDs (supporting nearly 300 printers, many driven to photo quality output), to be used alongside the Gimp-Print CUPS filters;
- TurboPrint (http://www.turboprint.com/) (http://www.turboprint.com/) (Shareware, non-Freee) supports roughly the same amount of printers in excellent quality;
- OMNI (http://www-124.ibm.com/developerworks/oss/linux/projects/omni/) (http://www-124.ibm.com/developerworks/oss/linux/projects/omni/) (LPGL, Free) is a package made by IBM, now containing support for more than 400 printers, stemming from the inheritance of IBM OS/2 KnowHow ported over to Linux (CUPS support is in a Beta-stage at present);
- HPIJS (http://hpinkjet.sourceforge.net/) (http://hpinkjet.sourceforge.net/) (BSD-style licnes, Free) supports around 120 of HP's own printers and is also providing excellent print quality now;
- Foomatic/cupsomatic (http://www.linuxprinting.org/) (http://www.linuxprinting.org/) (LPGL, Free) from Linuxprinting.org are providing PPDs for practically every Ghostscript filter known to the world, now usable with CUPS.

NOTE: the cupsomatic trick from Linuxprinting.org is working different from the other drivers. While the other drivers take the generic CUPS raster (produced by CUPS' own pstoraster PostScript RIP) as their input, cupsomatic "kidnaps" the PostScript inside CUPS, before RIP-ping, deviates it to an external Ghostscript installation (which now becomes the RIP) and gives it back to a CUPS backend once Ghostscript is finished. -- CUPS versions from 1.1.15 and later will provide their pstoraster PostScript RIP function again inside a system-wide Ghostscript installation rather than in "their own" pstoraster filter. (This CUPS-enabling Ghostscript version may be installed either as a patch to GNU or AFPL Ghostscript, or as a complete ESP Ghostscript package). However, this will not change the cupsomatic approach of guiding the printjob along a different path through the filtering system than the standard CUPS way...

Once you installed a printer inside CUPS with one of the recommended methods (the lpadmin command, the web browser interface or one of the available GUI wizards), you can use **cupsaddsmb** to share the printer via Samba. **cupsaddsmb** prepares the driver files for comfortable client download and installation upon their first contact with this printer share.

13.7.1. cupsaddsmb

The **cupsaddsmb** command copies the needed files for convenient Windows client installations from the previously prepared CUPS data directory to your [print$] share. Additionally, the PPD associated with this printer is copied from `/etc/cups/ppd/` to [print$].

```
root#  cupsaddsmb -U root infotec_IS2027
Password for root required to access localhost via SAMBA: [type in password ’secret’]
```

To share all printers and drivers, use the `-a` parameter instead of a printer name.

Probably you want to see what’s going on. Use the `-v` parameter to get a more verbose output:

Probably you want to see what’s going on. Use the `-v` parameter to get a more verbose output:

```
Note: The following line shave been wrapped so that information is not lost.

root#  cupsaddsmb -v -U root infotec_IS2027
    Password for root required to access localhost via SAMBA:
    Running command: smbclient //localhost/print\$ -N -U’root%secret’ -c ’mkdir W32X86;put
       /var/spool/cups/tmp/3cd1cc66376c0 W32X86/infotec_IS2027.PPD;put /usr/share/cups/drivers/
       ADOBEPS5.DLL W32X86/ADOBEPS5.DLL;put /usr/share/cups/drivers/ADOBEPSU.DLLr
       W32X86/ADOBEPSU.DLL;put /usr/share/cups/drivers/ADOBEPSU.HLP W32X86/ADOBEPSU.HLP’
    added interface ip=10.160.16.45 bcast=10.160.31.255 nmask=255.255.240.0
    added interface ip=192.168.182.1 bcast=192.168.182.255 nmask=255.255.255.0
    added interface ip=172.16.200.1 bcast=172.16.200.255 nmask=255.255.255.0
    Domain=[TUX-NET] OS=[Unix] Server=[Samba 2.2.3a.200204262025cvs]
    NT_STATUS_OBJECT_NAME_COLLISION making remote directory \W32X86
    putting file /var/spool/cups/tmp/3cd1cc66376c0 as \W32X86/infotec_IS2027.PPD (17394.6 kb/s)
      (average 17395.2 kb/s)
    putting file /usr/share/cups/drivers/ADOBEPS5.DLL as \W32X86/ADOBEPS5.DLL (10877.4 kb/s)
      (average 11343.0 kb/s)
    putting file /usr/share/cups/drivers/ADOBEPSU.DLL as \W32X86/ADOBEPSU.DLL (5095.2 kb/s)
      (average 9260.4 kb/s)
    putting file /usr/share/cups/drivers/ADOBEPSU.HLP as \W32X86/ADOBEPSU.HLP (8828.7 kb/s)
      (average 9247.1 kb/s)

    Running command: smbclient //localhost/print\$ -N -U’root%secret’ -c ’mkdir WIN40;put
      /var/spool/cups/tmp/3cd1cc66376c0 WIN40/infotec_IS2027.PPD;put
      /usr/share/cups/drivers/ADFONTS.MFM WIN40/ADFONTS.MFM;put
      /usr/share/cups/drivers/ADOBEPS4.DRV WIN40/ADOBEPS4.DRV;put
      /usr/share/cups/drivers/ADOBEPS4.HLP WIN40/ADOBEPS4.HLP;put
      /usr/share/cups/drivers/DEFPRTR2.PPD WIN40/DEFPRTR2.PPD;put
      /usr/share/cups/drivers/ICONLIB.DLL WIN40/ICONLIB.DLL;put
      /usr/share/cups/drivers/PSMON.DLL WIN40/PSMON.DLL;’
    added interface ip=10.160.16.45 bcast=10.160.31.255 nmask=255.255.240.0
    added interface ip=192.168.182.1 bcast=192.168.182.255 nmask=255.255.255.0
    added interface ip=172.16.200.1 bcast=172.16.200.255 nmask=255.255.255.0
    Domain=[TUX-NET] OS=[Unix] Server=[Samba 2.2.3a.200204262025cvs]
```

```
NT_STATUS_OBJECT_NAME_COLLISION making remote directory \WIN40
putting file /var/spool/cups/tmp/3cd1cc66376c0 as \WIN40/infotec_IS2027.PPD (26091.5 kb/s)
  (average 26092.8 kb/s)
putting file /usr/share/cups/drivers/ADFONTS.MFM as \WIN40/ADFONTS.MFM (11241.6 kb/s)
  (average 11812.9 kb/s)
putting file /usr/share/cups/drivers/ADOBEPS4.DRV as \WIN40/ADOBEPS4.DRV (16640.6 kb/s)
  (average 14679.3 kb/s)
putting file /usr/share/cups/drivers/ADOBEPS4.HLP as \WIN40/ADOBEPS4.HLP (11285.6 kb/s)
  (average 14281.5 kb/s)
putting file /usr/share/cups/drivers/DEFPRTR2.PPD as \WIN40/DEFPRTR2.PPD (823.5 kb/s)
  (average 12944.0 kb/s)
putting file /usr/share/cups/drivers/ICONLIB.DLL as \WIN40/ICONLIB.DLL (19226.2 kb/s)
  (average 13169.7 kb/s)
putting file /usr/share/cups/drivers/PSMON.DLL as \WIN40/PSMON.DLL (18666.1 kb/s)
  (average 13266.7 kb/s)

Running command: rpcclient localhost -N -U'root%secret' -c 'adddriver "Windows NT x86"
   "infotec_IS2027:ADOBEPS5.DLL:infotec_IS2027.PPD:ADOBEPSU.DLL:ADOBEPSU.HLP:NULL:RAW:NULL"'
cmd = adddriver "Windows NT x86" "infotec_IS2027:ADOBEPS5.DLL:infotec_IS2027.PPD:ADOBEPSU.DLL:
   ADOBEPSU.HLP:NULL:RAW:NULL"
Printer Driver infotec_IS2027 successfully installed.

Running command: rpcclient localhost -N -U'root%secret' -c 'adddriver "Windows 4.0"
   "infotec_IS2027:ADOBEPS4.DRV:infotec_IS2027.PPD:NULL:ADOBEPS4.HLP:PSMON.DLL:RAW:
   ADFONTS.MFM,DEFPRTR2.PPD,ICONLIB.DLL"'
cmd = adddriver "Windows 4.0" "infotec_IS2027:ADOBEPS4.DRV:infotec_IS2027.PPD:NULL:
   ADOBEPS4.HLP:PSMON.DLL:RAW:ADFONTS.MFM,DEFPRTR2.PPD,ICONLIB.DLL"
Printer Driver infotec_IS2027 successfully installed.

Running command: rpcclient localhost -N -U'root%secret'
   -c 'setdriver infotec_IS2027 infotec_IS2027'
cmd = setdriver infotec_IS2027 infotec_IS2027
Succesfully set infotec_IS2027 to driver infotec_IS2027.

root#
```

If you look closely, you'll discover your root password was transfered unencrypted over the wire, so beware! Also, if you look further her, you'll discover error messages like `NT_STATUS_OBJECT_NAME_COLLISION` in between. They occur, because the directories `WIN40` and `W32X86` already existed in the [print$] driver download share (from a previous driver installation). They are harmless here.

Now your printer is prepared for the clients to use. From a client, browse to the CUPS/Samba server, open the "Printers" share, right-click on this printer and select "Install..." or "Connect..." (depending on the Windows version you use). Now their should be a new printer in your client's local "Printers" folder, named (in my case) "infotec_IS2027 on kdebitshop"

NOTE: **cupsaddsmb** will only reliably work i with CUPS version 1.1.15 or higher and Samba from 2.2.4. If it doesn't work, or if the automatic printer driver download to the clients doesn't succeed, you can still manually install the CUPS printer PPD on top of the Adobe PostScript driver on clients and then point the client's printer queue to the Samba printer share for connection, should you desire to use the CUPS networked PostScript RIP functions.

13.8. The CUPS Filter Chains

The following diagrams reveal how CUPS handles print jobs.

```
#########################################################################
#
# CUPS in and of itself has this (general) filter chain (CAPITAL
# letters are FILE-FORMATS or MIME types, other are filters (this is
# true for pre-1.1.15 of pre-4.3 versions of CUPS and ESP PrintPro):
#
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#     somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#     pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT
#      |
#      |
#      V
#     pstoraster   # as shipped with CUPS, independent from any Ghostscipt
#      |           # installation on the system
#      |  (= "postscipt interpreter")
#      |
#      V
# APPLICATION/VND.CUPS-RASTER
#      |
#      |
#      V
#     rastertosomething  (f.e. Gimp-Print filters may be plugged in here)
#      |   (= "raster driver")
#      |
#      V
# SOMETHING-DEVICE-SPECIFIC
#      |
#      |
#      V
#     backend
#
#
# ESP PrintPro has some enhanced "rastertosomething" filters as compared to
# CUPS, and also a somewhat improved "pstoraster" filter.
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
#
```

```
##########################################################################

##########################################################################
#
# This is how "cupsomatic" comes into play:
# ==========================================
#
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#    somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#    pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT ----------------+
#      |                                          |
#      |                                          V
#      V                                         cupsomatic
#    pstoraster                                  (constructs complicated
#      |  (= "postscipt interpreter")            Ghostscript commandline
#      |                                         to let the file be
#      V                                         processed by a
# APPLICATION/VND.CUPS-RASTER                    "-sDEVICE=s.th."
#      |                                         call...)
#      |                                          |
#      V                                          |
#    rastertosomething                          V
#      |    (= "raster driver")     +-------------------------+
#      |                            | Ghostscript at work.... |
#      V                            |                         |
# SOMETHING-DEVICE-SPECIFIC         *-------------------------+
#      |                                          |
#      |                                          |
#      V                                          |
#    backend >------------------------------------+
#      |
#      |
#      V
#    THE PRINTER
#
#
# Note, that cupsomatic "kidnaps" the printfile after the
# "APPLICATION/VND.CUPS-POSTSCRPT" stage and deviates it through
# the CUPS-external, systemwide Ghostscript installation, bypassing the
# "pstoraster" filter (therefor also bypassing the CUPS-raster-drivers
# "rastertosomething", and hands the rasterized file directly to the CUPS
# backend...
#
```

```
# cupsomatic is not made by the CUPS developers. It is an independent
# contribution to printing development, made by people from
# Linuxprinting.org. (see also http://www.cups.org/cups-help.html)
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
#
#########################################################################

#########################################################################
#
# And this is how it works for ESP PrintPro from 4.3:
# ===================================================
#
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#     somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#     pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT
#      |
#      |
#      V
#     gsrip
#      |  (= "postscipt interpreter")
#      |
#      V
# APPLICATION/VND.CUPS-RASTER
#      |
#      |
#      V
#     rastertosomething  (f.e. Gimp-Print filters may be plugged in here)
#      |   (= "raster driver")
#      |
#      V
# SOMETHING-DEVICE-SPECIFIC
#      |
#      |
#      V
#     backend
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
#
#########################################################################

#########################################################################
```

```
#
# This is how "cupsomatic" would come into play with ESP PrintPro:
# ================================================================
#
#
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#    somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#    pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT ----------------+
#      |                                          |
#      |                                          V
#      V                                         cupsomatic
#    gsrip                                       (constructs complicated
#      |  (= "postscipt interpreter")            Ghostscript commandline
#      |                                         to let the file be
#      V                                         processed by a
# APPLICATION/VND.CUPS-RASTER                    "-sDEVICE=s.th."
#      |                                         call...)
#      |                                          |
#      V                                          |
#    rastertosomething                          V
#      |   (= "raster driver")      +-------------------------+
#      |                            | Ghostscript at work.... |
#      V                            |                         |
# SOMETHING-DEVICE-SPECIFIC        *-------------------------+
#      |                                          |
#      |                                          |
#      V                                          |
#    backend >------------------------------------+
#      |
#      |
#      V
#    THE PRINTER
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
#
#########################################################################

#########################################################################
#
# And this is how it works for CUPS from 1.1.15:
# ==============================================
#
```

```
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#     somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#     pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT-----+
#                                     |
#                  +------------------v------------------------------+
#                  | Ghostscript                                     |
#                  | at work...                                      |
#                  | (with                                           |
#                  | "-sDEVICE=cups")                                |
#                  |                                                 |
#                  |         (= "postscipt interpreter")             |
#                  |                                                 |
#                  +------------------v------------------------------+
#                                     |
#                                     |
# APPLICATION/VND.CUPS-RASTER >-------+
#      |
#      |
#      V
#     rastertosomething
#      |   (= "raster driver")
#      |
#      V
# SOMETHING-DEVICE-SPECIFIC
#      |
#      |
#      V
#     backend
#
#
# NOTE: since version 1.1.15 CUPS "outsourced" the pstoraster process to
#       Ghostscript. GNU Ghostscript needs to be patched to handle the
#       CUPS requirement; ESP Ghostscript has this builtin. In any case,
#       "gs -h" needs to show up a "cups" device. pstoraster is now a
#       calling an appropriate "gs -sDEVICE=cups..." commandline to do
#       the job. It will output "application/vnd.cup-raster", which will
#       be finally processed by a CUPS raster driver "rastertosomething"
#       Note the difference to "cupsomatic", which will *not* output
#       CUPS-raster, but a final version of the printfile, ready to be
#       sent to the printer. cupsomatic also doesn’t use the "cups"
#       devicemode in Ghostscript, but one of the classical devicemodes....
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
```

```
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
#
#########################################################################

#########################################################################
#
# And this is how it works for CUPS from 1.1.15, with cupsomatic included:
# ========================================================================
#
# SOMETHNG-FILEFORMAT
#      |
#      |
#      V
#     somethingtops
#      |
#      |
#      V
# APPLICATION/POSTSCRIPT
#      |
#      |
#      V
#     pstops
#      |
#      |
#      V
# APPLICATION/VND.CUPS-POSTSCRIPT-----+
#                                     |
#                  +------------------v------------------------------+
#                  | Ghostscript        . Ghostscript at work....    |
#                  | at work...         . (with "-sDEVICE=           |
#                  | (with              .            s.th."        |
#                  | "-sDEVICE=cups")   .                            |
#                  |                    .                            |
#                  | (CUPS standard)    .      (cupsomatic)          |
#                  |                    .                            |
#                  |          (= "postscript interpreter")           |
#                  |                    .                            |
#                  +------------------v--------------v---------------+
#                                     |              |
#                                     |              |
# APPLICATION/VND.CUPS-RASTER >-------+              |
#      |                                             |
#      |                                             |
#      V                                             |
#     rastertosomething                            |
#      |   (= "raster driver")                       |
#      |                                             |
#      V                                             |
# SOMETHING-DEVICE-SPECIFIC >------------------------+
#      |
#      |
#      V
#     backend
#
#
# NOTE: Gimp-Print and some other 3rd-Party-Filters (like TurboPrint) to
#       CUPS and ESP PrintPro plug-in where rastertosomething is noted.
```

```
#
############################################################################
```

13.9. CUPS Print Drivers and Devices

CUPS ships with good support for HP LaserJet type printers. You can install the driver as follows:

- lpadmin -p laserjet4plus -v parallel:/dev/lp0 -E -m laserjet.ppd

(The "-m" switch will retrieve the "laserjet.ppd" from the standard repository for not-yet-installed-PPDs, which CUPS typically stores in `/usr/share/cups/model`. Alternatively, you may use "-P /absolute/filesystem/path/to/where/there/is/PPD/your.ppd").

13.9.1. Further printing steps

Always also consult the database on linuxprinting.org for all recommendations about which driver is best used for each printer:

http://www.linuxprinting.org/printer_list.cgi

There select your model and click on "Show". You'll arrive at a page listing all drivers working with your model. There will always be *one* *recommended* one. Try this one first. In your case ("HP LaserJet 4 Plus"), you'll arrive here:

http://www.linuxprinting.org/show_printer.cgi?recnum=75104

The recommended driver is "ljet4". It has a link to the page for the ljet4 driver too:

http://www.linuxprinting.org/show_driver.cgi?driver=ljet4

On the driver's page, you'll find important and detailed info about how to use that driver within the various available spoolers. You can generate a PPD for CUPS. The PPD contains all the info about how to use your model and the driver; this is, once installed, working transparently for the user -- you'll only need to choose resolution, paper size etc. from the web-based menu or from the print dialog GUI or from the commandline...

On the driver's page, choose to use the "PPD-O-Matic" online PPD generator program. Select your model and click "Generate PPD file". When you safe the appearing ASCII text file, don't use "cut'n'past" (as it could possiblly corrupt line endings and tabs), but use "Save as..." in your browser's menu. Save it at "/some/path/on/your/filesystem/somewhere/my-name-for-my-printer.ppd"

Then install the printer:

```
"lpadmin -p laserjet4plus -v parallel:/dev/lp0 -E \
      -P /some/path/on/your/filesystem/somewhere/my-name-for-my-printer.ppd"
```

Note, that for all the "Foomatic-PPDs" from Linuxprinting.org, you also need a special "CUPS filter" named "cupsomatic". Get the latest version of "cupsomatic" from:

http://www.linuxprinting.org/cupsomatic

This needs to be copied to `/usr/lib/cups/filter/cupsomatic` and be made world executable. This filter is needed to read and act upon the specially encoded Foomatic comments, embedded in the printfile, which in turn are used to construct (transparently for you, the user) the complicated ghostscript command line needed for your printer/driver combo.

You can have a look at all the options for the Ghostscript commandline supported by your printer and the ljet4 driver by going to the section "Execution details", selecting your model (Laserjet 4 Plus) and clicking on "Show execution details". This will bring up this web page:

http://www.linuxprinting.org/execution.cgi?driver=ljet4&printer=75104&.submit=Show+execution+details

The ingenious thing is that the database is kept current. If there is a bug fix and an improvement somewhere in the database, you will always get the most current and stable and feature-rich driver by following the steps described above.

> **Note:** Till Kamppeter from MandrakeSoft is doing an excellent job here that too few people are aware of. (So if you use it often, please send him a note showing your appreciation).

The latest and greatest improvement now is support for "custom page sizes" for all those printers which support it.

"cupsomatic" is documented here:

http://www.linuxprinting.org/cups-doc.html

More printing tutorial info may be found here:

http://www.linuxprinting.org/kpfeifle/LinuxKongress2002/Tutorial/

Note, that *all* the Foomatic drivers listed on Linuxprinting.org (now approaching the "all-time high" number of 1.000 for the supported models) are using a special filtering chain involving Ghostscript, as described in this document.

Summary - You need:

A "foomatic+*`something`*" PPD is not enough to print with CUPS (but it is *one* important component)
The "cupsomatic" filter script (Perl) in `/usr/lib/cups/filters/`
Perl to make cupsomatic run
Ghostscript (because it is called and controlled by the PPD/cupsomatic combo in a way to fit your printermodel/driver combo.
Ghostscript *must*, depending on the driver/model, contain support for a certain "device" (as shown by "gs -h")

In the case of the "hpijs" driver, you need a Ghostscript version, which has "ijs" amongst its supported devices in "gs -h". In the case of "hpijs+foomatic", a valid ghostscript commandline would be reading like this:

```
gs -q -dBATCH -dPARANOIDSAFER -dQUIET -dNOPAUSE -sDEVICE=ijs       \
      -sIjsServer=hpijsPageSize -dDuplex=Duplex Model        \
      -rResolution,PS:MediaPosition=InputSlot -dIjsUseOutputFD \
      -sOutputFile=- -
```

> **Note:** Note, that with CUPS and the "hpijs+foomatic" PPD (plus Perl and cupsomatic) you don't need to remember this. You can choose the available print options thru a GUI print command (like "glp" from ESP's commercially supported PrintPro software, or KDE's "kprinter", or GNOME's "gtklp" or the independent "xpp") or the CUPS web interface via human-readable drop-down selection menus.

If you use "ESP Ghostscript" (also under the GPL, provided by Easy Software Products, the makers of CUPS, downloadable from http://www.cups.org/software.html, co-maintained by the developers of linuxprinting.org), you are guaranteed to have in use the most uptodate, bug-fixed, enhanced and stable version of a Free Ghostscript. It contains support for ~300 devices, whereas plain vanilla GNU Ghostscript 7.05 only has ~200.

If you print only one CUPS test page, from the web interface and when you try to print a windows test page, it acts like the job was never sent:

Can you print "standard" jobs from the CUPS machine?
Are the jobs from Windows visible in the Web interface on CUPS (http://localhost:631/)?
Most important: What kind of printer driver are you using on the Windows clients?

You can try to get a more detailed debugging info by setting "LogLevel debug" in `/etc/cups/cupsd.conf`, re-start cupsd and investigate `/var/log/cups/error_log` for the whereabouts of your Windows-originating printjobs:

what does the "auto-typing" line say? which is the "MIME type" CUPS thinks is arriving from the Windows clients?
are there "filter" available for this MIME type?
are there "filter rules" defined in "/etc/cups/mime.convs" for this MIME type?

13.10. Limiting the number of pages users can print

The feature you want is dependent on the real print subsystem you're using. Samba's part is always to receive the job files from the clients (filtered *or* unfiltered) and hand it over to this printing subsystem.

Of course one could "hack" things with one's own scripts.

But there is CUPS (Common Unix Printing System). CUPS supports "quotas". Quotas can be based on sizes of jobs or on the number of pages or both, and are spanning any time period you want.

This is an example command how root would set a print quota in CUPS, assuming an existing printer named "quotaprinter":

```
lpadmin -p quotaprinter -o job-quota-period=604800 -o job-k-limit=1024 \
     -o job-page-limit=100
```

This would limit every single user to print 100 pages or 1024 KB of data (whichever comes first) within the last 604.800 seconds (= 1 week).

For CUPS to count correctly, the printfile needs to pass the CUPS "pstops" filter, otherwise it uses a "dummy" count of "1". Some printfiles don't pass it (eg: image files) but then those are mostly 1 page jobs anyway. This also means, proprietary drivers for the target printer running on the client computers and CUPS/Samba then spooling these files as "raw" (i.e. leaving them untouched, not filtering them), will be counted as "1-pagers" too!

You need to send PostScript from the clients (i.e. run a PostScript driver there) for having the chance to get accounting done. If the printer is a non-PostScript model, you need to let CUPS do the job to convert the file to a print-ready format for the target printer. This will be working for currently ~1.000 different printer models, see

```
http://www.linuxprinting.org/printer_list.cgi
```

Before CUPS-1.1.16 your only option was to use the Adobe PostScript Driver on the Windows clients. The output of this driver was not always passed thru the "pstops" filter on the CUPS/Samba side, and therefor was not counted correctly (the reason is that it often --- depending on the "PPD" being used --- did write a "PJL"-header in front of the real PostScript which made CUPS to skip the pstops and go directy to the "pstoraster" stage).

From CUPS-1.1.16 onward you can use the "CUPS PostScript Driver for Windows NT/2K/XP clients" (it is tagged in the download area of http://www.cups.org/ as the "cups-samba-1.1.16.tar.gz" package). It is *not* working for Win9x/ME clients. But it:

it guarantees to not write an PJL-header
it guarantees to still read and support all PJL-options named in the driver PPD with its own means
it guarantees the file going thru the "pstops" filter on the CUPS/Samba server
it guarantees to page-count correctly the printfile

You can read more about the setup of this combination in the manpage for "cupsaddsmb" (only present with CUPS installed, only current with CUPS 1.1.16).

These are the items CUPS logs in the "page_log" for every single *page* of a job:

Printer name
User name
Job ID
Time of printing
the page number
the number of copies
a billing info string (optional)

Here is an extract of my CUPS server's page_log file to illustrate the format and included items:

```
infotec_IS2027 kurt 40 [22/Nov/2002:13:18:03 +0100] 1 2 #marketing infotec_IS2027
kurt 40 [22/Nov/2002:13:18:03 +0100] 2 2 #marketing infotec_IS2027 kurt 40
[22/Nov/2002:13:18:03 +0100] 3 2 #marketing infotec_IS2027 kurt 40
[22/Nov/2002:13:18:03 +0100] 4 2 #marketing infotec_IS2027 kurt 40
[22/Nov/2002:13:18:03 +0100] 5 2 #marketing infotec_IS2027 kurt 40
[22/Nov/2002:13:18:03 +0100] 6 2 #marketing
```

This was Job ID "40", printed on "infotec_IS2027" by user "kurt", a 6-page job printed in 2 copies and billed to "#marketing"...

What flaws or shortcomings are there?

the ones named above
CUPS really counts the job pages being *processsed in software* (going thru the "RIP") rather than the physical sheets successfully lea
all quotas are the same for all users (no flexibility to give the boss a higher quota than the clerk) no support for groups
no means to read out the current balance or "used-up" number of current quota
a user having used up 99 sheets of 100 quota will still be able to send and print a 1.000 sheet job
a user being denied a job because of a filled-up quota doesn't get a meaningful error message from CUPS other than "client-error-not-p

But this is the best system out there currently. And there are huge improvements under development:

page counting will go into the "backends" (these talk directly to the printer and will increase the count in sync with the actual printing p
quotas will be handled more flexibly
probably there will be support for users to inquire their "accounts" in advance
probably there will be support for some other tools around this topic

Other than the current stage of the CUPS development, I don't know any other ready-to-use tool which you could consider.

You can download the driver files from http://www.cups.org/software.html. It is a separate package from the CUPS base software files, tagged as "CUPS 1.1.16 Windows NT/2k/XP Printer Driver for SAMBA (tar.gz, 192k)". The filename to download is "cups-samba-1.1.16.tar.gz". Upon untar-/unzip-ping it will reveal the files:

```
cups-samba.install cups-samba.license cups-samba.readme cups-samba.remove
cups-samba.ss
```

These have been packaged with the ESP meta packager software "EPM". The *.install and *.remove files are simple shell script, which untars the *.ss (which is nothing else than a tar-archive) and puts its contents into `/usr/share/cups/drivers/`. Its contents are 3 files:

```
cupsdrvr.dll cupsui.dll cups.hlp
```

Caution

Due to a bug one CUPS release puts the `cups.hlp` into `/usr/share/drivers/` instead of `/usr/share/cups/drivers/`. To work around this, copy/move the file after running the "./cups-samba.install" script manually to the right place:

```
cp /usr/share/drivers/cups.hlp /usr/share/cups/drivers/
```

Note: This new CUPS PostScript driver is currently binary-only, but free no source code is provided (yet). The reason is this: it has been developed with the help of the Microsoft Driver Developer Kit (DDK) and compiled with Microsoft Visual Studio 6. It is not clear to the driver developers if they are allowed to distribute the whole of the source code as Free Software. However, they will likely release the "diff" in source code under the GPL, so anybody with a license of Visual Studio and a DDK will be able to compile for him/herself.

Once you have run the install script (and possibly manually moved the "cups.hlp" file to "/usr/share/cups/drivers/"), the driver is ready to be put into Samba's [print$] share (which often maps to "/etc/samba/drivers/" and contains a subdir tree with WIN40 and W32X86 branches), by running "cupsaddsmb" (see also "man cupsaddsmb" for CUPS 1.1.16). [Don't forget to put root into the smbpasswd file by running "smbpasswd" should you run this whole procedure for the first time.] Once the driver files are in the [print$] share, they are ready to be downloaded and installed by the Win NT/2k/XP clients.

Note: Win 9x/ME clients won't work with this driver. For these you'd still need to use the ADOBE*.* drivers as previously.

Note: It is not harming if you've still the ADOBE*.* driver files from previous installations in the "/usr/share/cups/drivers/" directory. The new cupsaddsmb (from 1.1.16) will automatically use the "newest" installed driver (which here then is the CUPS drivers).

Note: Should your Win clients have had the old ADOBE*.* files and the Adobe PostScript drivers installed, the download and installation of the new CUPS PostScript driver for Windows NT/2k/XP will fail at first.

It is not enough to "delete" the printer (as the driver files will still be kept by the clients and re-used if you try to re-install the printer). To really get rid of the Adobe driver files on the clients, open the "Printers" folder (possibly via "Start --> Settings --> Control Panel --> Printers"), right-click onto the folder background and select "Server Properties". A new dialog opens; select the "Drivers" tab; on the list select the driver you want to delete and click on the "Delete" button. (This will only work if there is no single printer left which uses that particular driver -- you need to "delete" all printers using this driver in the "Printers" folder first.)

Note: Once you have successfully downloaded the CUPS PostScript driver to a client, you can easily switch all printers to this one by proceeding as described elsewhere in the "Samba HOWTO Collection" to change a driver for an existing printer.

What are the benefits with the "CUPS PostScript driver for Windows NT/2k/XP" as compared to the Adobe drivers?

- no hassle with the Adobe EULA
- no hassle with the question "where do I get the ADOBE*.* driver files from?"
- the Adobe drivers (depending on the printer PPD associated with them) often put a PJL header in front of the core PostScript part of the print file (thus the file starts with "*1B%*-12345X" or "*escape%*-12345X" instead of "%!PS"). This leads to the CUPS daemon autotyping the arriving file as a print-ready file, not requiring a pass thru the "pstops" filter (to speak more technical, it is not regarded as the generic MIME type "application/postscript", but as the more special MIME type "application/cups.vnd-postscript"), which therefore also leads to the page accounting in "/var/log/cups/page_log" not receiving the exact mumber of pages; instead the dummy page number of "1" is logged in a standard setup)
- the Adobe driver has more options to "mis-configure" the PostScript generated by it (like setting it inadvertedly to "Optimize for Speed", instead of "Optimize for Portability", which could lead to CUPS being unable to process it)
- the CUPS PostScript driver output sent by Windows clients to the CUPS server will be guaranteed to be auto-typed as generic MIME type "application/postscript", thusly passing thru the CUPS "pstops" filter and logging the correct number of pages in the page_log for accounting and quota purposes
- the CUPS PostScript driver supports the sending of additional print options by the Win NT/2k/XP clients, such as naming the CUPS standard banner pages (or the custom ones, should they be installed at the time of driver download), using the CUPS "page-label" option, setting a job-priority and setting the scheduled time of printing (with the option to support additional useful IPP job attributes in the future).
- the CUPS PostScript driver supports the inclusion of the new "*cupsJobTicket" comments at the beginnig of the PostScript file (which could be used in the future for all sort of beneficial extensions on the CUPS side, but which will not disturb any other application as those will regard it as a comment and simply ignore it).
- the CUPS PostScript driver will be the heart of the fully fledged CUPS IPP client for Windows NT/2k/XP to be released soon (probably alongside the first Beta release for CUPS 1.2).

13.11. Advanced Postscript Printing from MS Windows

Let the Windows Clients use a PostScript driver to deliver poistscript to the samba print server (just like any Linux or Unix Client would also use PostScript to send to the server)

Make the Unix printing subsystem to which Samba sends the job convert the incoming PostScript files to the native print format of the target printers (would be PCL if you have an HP printer)

Now if you are afraid that this would just mean using a *Generic* PostScript driver for the clients that has no Simplex/Duplex selection, and no paper tray choice, but you need them to be able to set up print jobs, with all the bells and whistles of your printers:-

Not possible with traditional spooling systems
But perfectly supported by CUPS (which uses "PPD" files to describe how to control the print options for PostScript and non-PostScrip

CUPS PPDs are working perfectly on Windows clients who use Adobe PostScript drivers (or the new CUPS PostScript driver for Windows NT/2K/XP). Clients can use them to setup the job to their liking and CUPS will use the received job options to make the (PCL-, ESC/P- or PostScript-) printer behave as required.

If you want to have the additional benefit of page count logging and accounting then the CUPS PostScript driver is the best choice (better than the Adobe one).

If you want to make the drivers downloadable for the clients then "cupsaddsmb" is your friend. It will setup the [print$] share on the Samba host to be ready to serve the clients for a "point and print" driver installation.

Warning

What strings are attached?

There are some. But, given the sheer CPU power you can buy nowadays, these can be overcome easily. The strings:

Well, if the CUPS/Samba side will have to print to many printers serving many users, you probably will need to set up a second server (which can do automatic load balancing with the first one, plus a degree of fail-over mechanism). Converting the incoming PostScript jobs, "interpreting" them for non-PostScript printers, amounts to the work of a "RIP" (Raster Image Processor) done in software. This requires more CPU and RAM than for the mere "raw spooling" task your current setup is solving. It all depends on the avarage and peak printing load the server should be able to handle.

13.12. Auto-Deletion of CUPS spool files

Samba print files pass thru two "spool" directories. One the incoming directory managed by Samba, (set eg: in the **path = /var/spool/samba** directive in the [printers] section of `smb.conf`). Second is the spool directory of your UNIX print subsystem. For CUPS it is normally "/var/spool/cups/", as set by the cupsd.conf directive "RequestRoot /var/spool/cups".

I am not sure, which one of your directories keeps the files. From what you say, it is most likely the Samba part.

For the CUPS part, you may want to consult:

http://localhost:631/sam.html#PreserveJobFiles
http://localhost:631/sam.html#PreserveJobHistory
http://localhost:631/sam.html#MaxJobs

There are the settings described for your CUPS daemon, which could lead to completed job files not being deleted.

"PreserveJobHistory Yes" -- keeps some details of jobs in cupsd's mind (well it keeps the "c12345", "c12346" etc. files in the CUPS spool directory, which do a similar job as the old-fashioned BSD-LPD control files). This is set to "Yes" as a default.

"PreserveJobFiles Yes" -- keeps the job files themselves in cupsd's mind (well it keeps the "d12345", "d12346" etc. files in the CUPS spool directory...). This is set to "No" as the CUPS default.

"MaxJobs 500" -- this directive controls the maximum number of jobs that are kept in memory. Once the number of jobs reaches the limit, the oldest completed job is automatically purged from the system to make room for the new one. If all of the known jobs are still pending or active then the new job will be rejected. Setting the maximum to 0 disables this functionality. The default setting is 0.

(There are also additional settings for "MaxJobsPerUser" and "MaxJobsPerPrinter"...)

For everything to work as announced, you need to have three things:

a Samba-smbd which is compiled against "libcups" (Check on Linux by running **`` ldd `which smbd` ``**)
a Samba-`smb.conf` setting of **printing = cups**
another Samba-`smb.conf` setting of **printcap = cups**

> **Note:** Note, that in this case all other manually set printing-related commands (like "print command", "lpq command", "lprm command", "lppause command" or "lpresume command") are ignored and they should normally have no influence what-so-ever on your printing.

If you want to do things manually, replace the "printing = cups" by "printing = bsd". Then your manually set commands may work (haven't tested this), and a "print command = lp -d %P %s; rm %s" may do what you need.

You forgot to mention the CUPS version you're using. If you did set things up as described in the man pages, then the Samba spool files should be deleted. Otherwise it may be a bug. On the CUPS side, you can control the behaviour as described above.

If you have more problems, post the output of these commands:

```
grep -v ^# /etc/cups/cupsd.conf | grep -v ^$ grep -v ^# /etc/samba/smb.conf
| grep -v ^$ | grep -v "^;"
```

(adapt paths as needed). These commands sanitize the files and cut out the empty lines and lines with comments, providing the "naked settings" in a compact way.

Chapter 14. Unified Logons between Windows NT and UNIX using Winbind

14.1. Abstract

Integration of UNIX and Microsoft Windows NT through a unified logon has been considered a "holy grail" in heterogeneous computing environments for a long time. We present *winbind*, a component of the Samba suite of programs as a solution to the unified logon problem. Winbind uses a UNIX implementation of Microsoft RPC calls, Pluggable Authentication Modules, and the Name Service Switch to allow Windows NT domain users to appear and operate as UNIX users on a UNIX machine. This paper describes the winbind system, explaining the functionality it provides, how it is configured, and how it works internally.

14.2. Introduction

It is well known that UNIX and Microsoft Windows NT have different models for representing user and group information and use different technologies for implementing them. This fact has made it difficult to integrate the two systems in a satisfactory manner.

One common solution in use today has been to create identically named user accounts on both the UNIX and Windows systems and use the Samba suite of programs to provide file and print services between the two. This solution is far from perfect however, as adding and deleting users on both sets of machines becomes a chore and two sets of passwords are required both of which can lead to synchronization problems between the UNIX and Windows systems and confusion for users.

We divide the unified logon problem for UNIX machines into three smaller problems:

- Obtaining Windows NT user and group information
- Authenticating Windows NT users
- Password changing for Windows NT users

Ideally, a prospective solution to the unified logon problem would satisfy all the above components without duplication of information on the UNIX machines and without creating additional tasks for the system administrator when maintaining users and groups on either system. The winbind system provides a simple and elegant solution to all three components of the unified logon problem.

14.3. What Winbind Provides

Winbind unifies UNIX and Windows NT account management by allowing a UNIX box to become a full member of a NT domain. Once this is done the UNIX box will see NT users and groups as if they were native UNIX users and groups, allowing the NT domain to be used in much the same manner that NIS+ is used within UNIX-only environments.

The end result is that whenever any program on the UNIX machine asks the operating system to lookup a user or group name, the query will be resolved by asking the NT domain controller for the specified domain to do the lookup. Because Winbind hooks into the operating system at a low level (via the NSS name resolution modules in the C library) this redirection to the NT domain controller is completely transparent.

Users on the UNIX machine can then use NT user and group names as they would use "native" UNIX names. They can chown files so that they are owned by NT domain users or even login to the UNIX machine and run a UNIX X-Window session as a domain user.

The only obvious indication that Winbind is being used is that user and group names take the form DOMAIN\user and DOMAIN\group. This is necessary as it allows Winbind to determine that redirection to a domain controller is wanted for a particular lookup and which trusted domain is being referenced.

Additionally, Winbind provides an authentication service that hooks into the Pluggable Authentication Modules (PAM) system to provide authentication via a NT domain to any PAM enabled applications. This capability solves the problem of synchronizing passwords between systems since all passwords are stored in a single location (on the domain controller).

14.3.1. Target Uses

Winbind is targeted at organizations that have an existing NT based domain infrastructure into which they wish to put UNIX workstations or servers. Winbind will allow these organizations to deploy UNIX workstations without having to maintain a separate account infrastructure. This greatly simplifies the administrative overhead of deploying UNIX workstations into a NT based organization.

Another interesting way in which we expect Winbind to be used is as a central part of UNIX based appliances. Appliances that provide file and print services to Microsoft based networks will be able to use Winbind to provide seamless integration of the appliance into the domain.

14.4. How Winbind Works

The winbind system is designed around a client/server architecture. A long running **winbindd** daemon listens on a UNIX domain socket waiting for requests to arrive. These requests are generated by the NSS and PAM clients and processed sequentially.

The technologies used to implement winbind are described in detail below.

14.4.1. Microsoft Remote Procedure Calls

Over the last few years, efforts have been underway by various Samba Team members to decode various aspects of the Microsoft Remote Procedure Call (MSRPC) system. This system is used for most network related operations between Windows NT machines including remote management, user authentication and print spooling. Although initially this work was done to aid the implementation of Primary Domain Controller (PDC) functionality in Samba, it has also yielded a body of code which can be used for other purposes.

Winbind uses various MSRPC calls to enumerate domain users and groups and to obtain detailed information about individual users or groups. Other MSRPC calls can be used to authenticate NT domain users and to change user passwords. By directly querying a Windows PDC for user and group information, winbind maps the NT account information onto UNIX user and group names.

14.4.2. Microsoft Active Directory Services

Since late 2001, Samba has gained the ability to interact with Microsoft Windows 2000 using its 'Native Mode' protocols, rather than the NT4 RPC services. Using LDAP and Kerberos, a domain member running winbind can enumerate users and groups in exactly the same way as a Win2k client would, and in so doing provide a much more efficient and effective winbind implementation.

14.4.3. Name Service Switch

The Name Service Switch, or NSS, is a feature that is present in many UNIX operating systems. It allows system information such as hostnames, mail aliases and user information to be resolved from different sources. For example, a standalone UNIX workstation may resolve system information from a series of flat files stored on the local filesystem. A networked workstation may first attempt to resolve system information from local files, and then consult a NIS database for user information or a DNS server for hostname information.

The NSS application programming interface allows winbind to present itself as a source of system information when resolving UNIX usernames and groups. Winbind uses this interface, and information obtained from a Windows NT server using MSRPC calls to provide a new source of account enumeration. Using standard UNIX library calls, one can enumerate the users and groups on a UNIX machine running winbind and see all users and groups in a NT domain plus any trusted domain as though they were local users and groups.

The primary control file for NSS is `/etc/nsswitch.conf`. When a UNIX application makes a request to do a lookup the C library looks in `/etc/nsswitch.conf` for a line which matches the service type being requested, for example the "passwd" service type is used when user or group names are looked up. This config line species which implementations of that service should be tried and in what order. If the passwd config line is:

passwd: files example

then the C library will first load a module called `/lib/libnss_files.so` followed by the module `/lib/libnss_example.so`. The C library will dynamically load each of these modules in turn and call resolver functions within the modules to try to resolve the request. Once the request is resolved the C library returns the result to the application.

This NSS interface provides a very easy way for Winbind to hook into the operating system. All that needs to be done is to put `libnss_winbind.so` in `/lib/` then add "winbind" into `/etc/nsswitch.conf` at the appropriate place. The C library will then call Winbind to resolve user and group names.

14.4.4. Pluggable Authentication Modules

Pluggable Authentication Modules, also known as PAM, is a system for abstracting authentication and authorization technologies. With a PAM module it is possible to specify different authentication methods for different system applications without having to recompile these applications. PAM is also useful for implementing a particular policy for authorization. For example, a system administrator may only allow console logins from users stored in the local password file but only allow users resolved from a NIS database to log in over the network.

Winbind uses the authentication management and password management PAM interface to integrate Windows NT users into a UNIX system. This allows Windows NT users to log in to a UNIX machine and be authenticated against a suitable Primary Domain Controller. These users can also change their passwords and have this change take effect directly on the Primary Domain Controller.

PAM is configured by providing control files in the directory `/etc/pam.d/` for each of the services that require authentication. When an authentication request is made by an application the PAM code in the C library looks up this control file to determine what modules to load to do the authentication check and in what order. This interface makes adding a new authentication service for Winbind very easy, all that needs to be done is that the `pam_winbind.so` module is copied to `/lib/security/` and the PAM control files for relevant services are updated to allow authentication via winbind. See the PAM documentation for more details.

14.4.5. User and Group ID Allocation

When a user or group is created under Windows NT is it allocated a numerical relative identifier (RID). This is slightly different to UNIX which has a range of numbers that are used to identify users, and the same range in which to identify groups. It is winbind's job to convert RIDs to UNIX id numbers and vice versa. When winbind is configured it is given part of the UNIX user id space and a part of the UNIX group id space in which to store Windows NT users and groups. If a Windows NT user is resolved for the first time, it is allocated the next UNIX id from the range. The same process applies for Windows NT groups. Over time, winbind will have mapped all Windows NT users and groups to UNIX user ids and group ids.

The results of this mapping are stored persistently in an ID mapping database held in a tdb database). This ensures that RIDs are mapped to UNIX IDs in a consistent way.

14.4.6. Result Caching

An active system can generate a lot of user and group name lookups. To reduce the network cost of these lookups winbind uses a caching scheme based on the SAM sequence number supplied by NT domain controllers. User or group information returned by a PDC is cached by winbind along with a sequence number also returned by the PDC. This sequence number is incremented by Windows NT whenever any user or group information is modified. If a cached entry has expired, the sequence number is requested from the PDC and compared against the sequence number of the cached entry. If the sequence numbers do not match, then the cached information is discarded and up to date information is requested directly from the PDC.

14.5. Installation and Configuration

Many thanks to John Trostel jtrostel@snapserver.com (mailto:jtrostel@snapserver.com) for providing the HOWTO for this section.

This HOWTO describes how to get winbind services up and running to control access and authenticate users on your Linux box using the winbind services which come with SAMBA 2.2.2.

14.5.1. Introduction

This HOWTO describes the procedures used to get winbind up and running on my RedHat 7.1 system. Winbind is capable of providing access and authentication control for Windows Domain users through an NT or Win2K PDC for 'regular' services, such as telnet a nd ftp, as well for SAMBA services.

This HOWTO has been written from a 'RedHat-centric' perspective, so if you are using another distribution, you may have to modify the instructions somewhat to fit the way your distribution works.

- *Why should I to this?*

  This allows the SAMBA administrator to rely on the authentication mechanisms on the NT/Win2K PDC for the authentication of domain members. NT/Win2K users no longer need to have separate accounts on the SAMBA server.

- *Who should be reading this document?*

  This HOWTO is designed for system administrators. If you are implementing SAMBA on a file server and wish to (fairly easily) integrate existing NT/Win2K users from your PDC onto the SAMBA server, this HOWTO is for you. That said, I am no NT or PAM expert, so you may find a better or easier way to accomplish these tasks.

14.5.2. Requirements

If you have a samba configuration file that you are currently using... *BACK IT UP!* If your system already uses PAM, *back up the `/etc/pam.d` directory contents!* If you haven't already made a boot disk, *MAKE ONE NOW!*

Messing with the pam configuration files can make it nearly impossible to log in to yourmachine. That's why you want to be able to boot back into your machine in single user mode and restore your `/etc/pam.d` back to the original state they were in if you get frustrated with the way things are going. ;-)

The latest version of SAMBA (version 3.0 as of this writing), now includes a functioning winbindd daemon. Please refer to the main SAMBA web page (http://samba.org/) or, better yet, your closest SAMBA mirror site for instructions on downloading the source code.

To allow Domain users the ability to access SAMBA shares and files, as well as potentially other services provided by your SAMBA machine, PAM (pluggable authentication modules) must be setup properly on your machine. In order to compile the winbind modules, you should have at least the pam libraries resident on your system. For recent RedHat systems (7.1, for instance), that means `pam-0.74-22`. For best results, it is helpful to also install the development packages in `pam-devel-0.74-22`.

14.5.3. Testing Things Out

Before starting, it is probably best to kill off all the SAMBA related daemons running on your server. Kill off all **smbd**, **nmbd**, and **winbindd** processes that may be running. To use PAM, you will want to make sure that you have the standard PAM package (for RedHat) which supplies the `/etc/pam.d` directory structure, including the pam modules are used by pam-aware services, several pam libraries, and the `/usr/doc` and `/usr/man` entries for pam. Winbind built better in SAMBA if the pam-devel package was also installed. This package includes the header files needed to compile pam-aware applications. For instance, my RedHat system has both `pam-0.74-22` and `pam-devel-0.74-22` RPMs installed.

14.5.3.1. Configure and compile SAMBA

The configuration and compilation of SAMBA is pretty straightforward. The first three steps may not be necessary depending upon whether or not you have previously built the Samba binaries.

```
root# autoconf
root# make clean
root# rm config.cache
root# ./configure
root# make
root# make install
```

This will, by default, install SAMBA in `/usr/local/samba`. See the main SAMBA documentation if you want to install SAMBA somewhere else. It will also build the winbindd executable and libraries.

14.5.3.2. Configure `nsswitch.conf` and the winbind libraries

The libraries needed to run the **winbindd** daemon through nsswitch need to be copied to their proper locations, so

```
root# cp ../samba/source/nsswitch/libnss_winbind.so /lib
```

I also found it necessary to make the following symbolic link:

```
root# ln -s /lib/libnss_winbind.so /lib/libnss_winbind.so.2
```

And, in the case of Sun solaris:

`root#` **ln -s /usr/lib/libnss_winbind.so /usr/lib/libnss_winbind.so.1** `root#` **ln -s /usr/lib/libnss_winbind.so /usr/lib/nss_winbind.so.1** `root#` **ln -s /usr/lib/libnss_winbind.so /usr/lib/nss_winbind.so.2**

Now, as root you need to edit `/etc/nsswitch.conf` to allow user and group entries to be visible from the **winbindd** daemon. My `/etc/nsswitch.conf` file look like this after editing:

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

The libraries needed by the winbind daemon will be automatically entered into the **ldconfig** cache the next time your system reboots, but it is faster (and you don't need to reboot) if you do it manually:

`root#` **/sbin/ldconfig -v | grep winbind**

This makes `libnss_winbind` available to winbindd and echos back a check to you.

14.5.3.3. Configure smb.conf

Several parameters are needed in the smb.conf file to control the behavior of **winbindd**. Configure `smb.conf` These are described in more detail in the winbindd(8) (winbindd.8.html) man page. My `smb.conf` file was modified to include the following entries in the [global] section:

```
[global]
     <...>
     # separate domain and username with '+', like DOMAIN+username
     winbind separator (winbindd.8.html#WINBINDSEPARATOR) = +
     # use uids from 10000 to 20000 for domain users
     winbind uid (winbindd.8.html#WINBINDUID) = 10000-20000
     # use gids from 10000 to 20000 for domain groups
     winbind gid (winbindd.8.html#WINBINDGID) = 10000-20000
     # allow enumeration of winbind users and groups
     winbind enum users (winbindd.8.html#WINBINDENUMUSERS) = yes
     winbind enum groups (winbindd.8.html#WINBINDENUMGROUP) = yes
     # give winbind users a real shell (only needed if they have telnet access)
     template homedir (winbindd.8.html#TEMPLATEHOMEDIR) = /home/winnt/%D/%U
     template shell (winbindd.8.html#TEMPLATESHELL) = /bin/bash
```

14.5.3.4. Join the SAMBA server to the PDC domain

Enter the following command to make the SAMBA server join the PDC domain, where *`DOMAIN`* is the name of your Windows domain and *`Administrator`* is a domain user who has administrative privileges in the domain.

```
root# /usr/local/samba/bin/net join -S PDC -U Administrator
```

The proper response to the command should be: "Joined the domain *`DOMAIN`*" where *`DOMAIN`* is your DOMAIN name.

14.5.3.5. Start up the winbindd daemon and test it!

Eventually, you will want to modify your smb startup script to automatically invoke the winbindd daemon when the other parts of SAMBA start, but it is possible to test out just the winbind portion first. To start up winbind services, enter the following command as root:

```
root# /usr/local/samba/bin/winbindd
```

Winbindd can now also run in 'dual daemon mode'. This will make it run as 2 processes. The first will answer all requests from the cache, thus making responses to clients faster. The other will update the cache for the query that the first has just responded. Advantage of this is that responses stay accurate and are faster. You can enable dual daemon mode by adding '-B' to the commandline:

```
root# /usr/local/samba/bin/winbindd -B
```

I'm always paranoid and like to make sure the daemon is really running...

```
root# ps -ae | grep winbindd
```

This command should produce output like this, if the daemon is running

```
3025 ? 00:00:00 winbindd
```

Now... for the real test, try to get some information about the users on your PDC

```
root# /usr/local/samba/bin/wbinfo -u
```

This should echo back a list of users on your Windows users on your PDC. For example, I get the following response:

```
CEO+Administrator
CEO+burdell
```

```
CEO+Guest
CEO+jt-ad
CEO+krbtgt
CEO+TsInternetUser
```

Obviously, I have named my domain 'CEO' and my *`winbind separator`* is '+'.

You can do the same sort of thing to get group information from the PDC:

```
root# /usr/local/samba/bin/wbinfo -g
CEO+Domain Admins
CEO+Domain Users
CEO+Domain Guests
CEO+Domain Computers
CEO+Domain Controllers
CEO+Cert Publishers
CEO+Schema Admins
CEO+Enterprise Admins
CEO+Group Policy Creator Owners
```

The function 'getent' can now be used to get unified lists of both local and PDC users and groups. Try the following command:

```
root# getent passwd
```

You should get a list that looks like your `/etc/passwd` list followed by the domain users with their new uids, gids, home directories and default shells.

The same thing can be done for groups with the command

```
root# getent group
```

14.5.3.6. Fix the init.d startup scripts

14.5.3.6.1. Linux

The **winbindd** daemon needs to start up after the **smbd** and **nmbd** daemons are running. To accomplish this task, you need to modify the startup scripts of your system. They are located at `/etc/init.d/smb` in RedHat and `/etc/init.d/samba` in Debian. script to add commands to invoke this daemon in the proper sequence. My startup script starts up **smbd**, **nmbd**, and **winbindd** from the `/usr/local/samba/bin` directory directly. The 'start' function in the script looks like this:

```
start() {
        KIND="SMB"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/bin/smbd $SMBDOPTIONS
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/bin/nmbd $NMBDOPTIONS
        RETVAL2=$?
        echo
        KIND="Winbind"
        echo -n $"Starting $KIND services: "
        daemon /usr/local/samba/bin/winbindd
        RETVAL3=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 -a $RETVAL3 -eq 0 ] && touch /var/lock/subsys/smb || \
           RETVAL=1
        return $RETVAL
}
```

If you would like to run winbindd in dual daemon mode, replace the line

```
        daemon /usr/local/samba/bin/winbindd
```

in the example above with:

```
        daemon /usr/local/samba/bin/winbindd -B
```

.

The 'stop' function has a corresponding entry to shut down the services and looks like this:

```
stop() {
        KIND="SMB"
        echo -n $"Shutting down $KIND services: "
        killproc smbd
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Shutting down $KIND services: "
        killproc nmbd
        RETVAL2=$?
        echo
        KIND="Winbind"
        echo -n $"Shutting down $KIND services: "
        killproc winbindd
        RETVAL3=$?
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 -a $RETVAL3 -eq 0 ] && rm -f /var/lock/subsys/smb
        echo ""
        return $RETVAL
}
```

14.5.3.6.2. Solaris

On solaris, you need to modify the `/etc/init.d/samba.server` startup script. It usually only starts smbd and nmbd but should now start winbindd too. If you have samba installed in `/usr/local/samba/bin`, the file could contains something like this:

```
##
## samba.server
##

if [ ! -d /usr/bin ]
then                    # /usr not mounted
        exit
fi

killproc() {            # kill the named process(es)
        pid=`/usr/bin/ps -e |
             /usr/bin/grep -w $1 |
             /usr/bin/sed -e ’s/^  *//’ -e ’s/ .*//’`
        [ "$pid" != "" ] && kill $pid
}

# Start/stop processes required for samba server

case "$1" in

’start’)
#
# Edit these lines to suit your installation (paths, workgroup, host)
#
echo Starting SMBD
   /usr/local/samba/bin/smbd -D -s \
/usr/local/samba/smb.conf

echo Starting NMBD
   /usr/local/samba/bin/nmbd -D -l \
/usr/local/samba/var/log -s /usr/local/samba/smb.conf

echo Starting Winbind Daemon
   /usr/local/samba/bin/winbindd
   ;;

’stop’)
   killproc nmbd
   killproc smbd
   killproc winbindd
   ;;

*)
   echo "Usage: /etc/init.d/samba.server { start | stop }"
   ;;
esac
```

Again, if you would like to run samba in dual daemon mode, replace

```
/usr/local/samba/bin/winbindd
```

in the script above with:

```
/usr/local/samba/bin/winbindd -B
```

14.5.3.6.3. Restarting

If you restart the **smbd**, **nmbd**, and **winbindd** daemons at this point, you should be able to connect to the samba server as a domain member just as if you were a local user.

14.5.3.7. Configure Winbind and PAM

If you have made it this far, you know that winbindd and samba are working together. If you want to use winbind to provide authentication for other services, keep reading. The pam configuration files need to be altered in this step. (Did you remember to make backups of your original `/etc/pam.d` files? If not, do it now.)

You will need a pam module to use winbindd with these other services. This module will be compiled in the `../source/nsswitch` directory by invoking the command

```
root# make nsswitch/pam_winbind.so
```

from the `../source` directory. The `pam_winbind.so` file should be copied to the location of your other pam security modules. On my RedHat system, this was the `/lib/security` directory. On Solaris, the pam security modules reside in `/usr/lib/security`.

```
root# cp ../samba/source/nsswitch/pam_winbind.so /lib/security
```

14.5.3.7.1. Linux/FreeBSD-specific PAM configuration

The `/etc/pam.d/samba` file does not need to be changed. I just left this fileas it was:

```
auth    required        /lib/security/pam_stack.so service=system-auth
account required        /lib/security/pam_stack.so service=system-auth
```

The other services that I modified to allow the use of winbind as an authentication service were the normal login on the console (or a terminal session), telnet logins, and ftp service. In order to enable these services, you may first need to change the entries in `/etc/xinetd.d` (or `/etc/inetd.conf`). RedHat 7.1 uses the new xinetd.d structure, in this case you need to change the lines in `/etc/xinetd.d/telnet` and `/etc/xinetd.d/wu-ftp` from

```
enable = no
```

to

```
enable = yes
```

For ftp services to work properly, you will also need to either have individual directories for the domain users already present on the server, or change the home directory template to a general directory for all domain users. These can be easily set using the `smb.conf` global entry **template homedir**.

The `/etc/pam.d/ftp` file can be changed to allow winbind ftp access in a manner similar to the samba file. My `/etc/pam.d/ftp` file was changed to look like this:

```
auth       required     /lib/security/pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=s
auth       sufficient   /lib/security/pam_winbind.so
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_shells.so
account    sufficient   /lib/security/pam_winbind.so
account    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

The `/etc/pam.d/login` file can be changed nearly the same way. It now looks like this:

```
auth       required     /lib/security/pam_securetty.so
auth       sufficient   /lib/security/pam_winbind.so
auth       sufficient   /lib/security/pam_unix.so use_first_pass
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    sufficient   /lib/security/pam_winbind.so
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    optional     /lib/security/pam_console.so
```

In this case, I added the **auth sufficient /lib/security/pam_winbind.so** lines as before, but also added the **required pam_securetty.so** above it, to disallow root logins over the network. I also added a **sufficient /lib/security/pam_unix.so use_first_pass** line after the **winbind.so** line to get rid of annoying double prompts for passwords.

14.5.3.7.2. Solaris-specific configuration

The /etc/pam.conf needs to be changed. I changed this file so that my Domain users can logon both locally as well as telnet.The following are the changes that I made.You can customize the pam.conf file as per your requirements,but be sure of those changes because in the worst case it will leave your system nearly impossible to boot.

```
#
#ident "@(#)pam.conf 1.14 99/09/16 SMI"
#
# Copyright (c) 1996-1999, Sun Microsystems, Inc.
# All Rights Reserved.
#
# PAM configuration
#
# Authentication management
#
login   auth required   /usr/lib/security/pam_winbind.so
login auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
login auth required  /usr/lib/security/$ISA/pam_dial_auth.so.1 try_first_pass
#
rlogin  auth sufficient /usr/lib/security/pam_winbind.so
rlogin  auth sufficient /usr/lib/security/$ISA/pam_rhosts_auth.so.1
rlogin auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
dtlogin auth sufficient /usr/lib/security/pam_winbind.so
dtlogin auth required  /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
rsh auth required /usr/lib/security/$ISA/pam_rhosts_auth.so.1
other   auth sufficient /usr/lib/security/pam_winbind.so
other auth required /usr/lib/security/$ISA/pam_unix.so.1 try_first_pass
#
# Account management
#
login   account sufficient      /usr/lib/security/pam_winbind.so
login account requisite /usr/lib/security/$ISA/pam_roles.so.1
login account required /usr/lib/security/$ISA/pam_unix.so.1
#
dtlogin account sufficient      /usr/lib/security/pam_winbind.so
dtlogin account requisite /usr/lib/security/$ISA/pam_roles.so.1
dtlogin account required /usr/lib/security/$ISA/pam_unix.so.1
#
other   account sufficient      /usr/lib/security/pam_winbind.so
other account requisite /usr/lib/security/$ISA/pam_roles.so.1
other account required /usr/lib/security/$ISA/pam_unix.so.1
#
# Session management
#
```

```
other session required /usr/lib/security/$ISA/pam_unix.so.1
#
# Password management
#
#other   password sufficient     /usr/lib/security/pam_winbind.so
other password required /usr/lib/security/$ISA/pam_unix.so.1
dtsession auth required /usr/lib/security/$ISA/pam_unix.so.1
#
# Support for Kerberos V5 authentication (uncomment to use Kerberos)
#
#rlogin auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#login auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#dtlogin auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#other auth optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
#dtlogin account optional /usr/lib/security/$ISA/pam_krb5.so.1
#other account optional /usr/lib/security/$ISA/pam_krb5.so.1
#other session optional /usr/lib/security/$ISA/pam_krb5.so.1
#other password optional /usr/lib/security/$ISA/pam_krb5.so.1 try_first_pass
```

I also added a try_first_pass line after the winbind.so line to get rid of annoying double prompts for passwords.

Now restart your Samba and try connecting through your application that you configured in the pam.conf.

14.6. Limitations

Winbind has a number of limitations in its current released version that we hope to overcome in future releases:

- Winbind is currently only available for the Linux, Solaris and IRIX operating systems, although ports to other operating systems are certainly possible. For such ports to be feasible, we require the C library of the target operating system to support the Name Service Switch and Pluggable Authentication Modules systems. This is becoming more common as NSS and PAM gain support among UNIX vendors.
- The mappings of Windows NT RIDs to UNIX ids is not made algorithmically and depends on the order in which unmapped users or groups are seen by winbind. It may be difficult to recover the mappings of rid to UNIX id mapping if the file containing this information is corrupted or destroyed.
- Currently the winbind PAM module does not take into account possible workstation and logon time restrictions that may be been set for Windows NT users, this is instead up to the PDC to enforce.

14.7. Conclusion

The winbind system, through the use of the Name Service Switch, Pluggable Authentication Modules, and appropriate Microsoft RPC calls have allowed us to provide seamless integration of Microsoft Windows NT domain users on a UNIX system. The result is a great reduction in the administrative cost of running a mixed UNIX and NT network.

Chapter 15. Advanced Network Manangement

This section attempts to document peripheral issues that are of great importance to network administrators who want to improve network resource access control, to automate the user environment, and to make their lives a little easier.

15.1. Configuring Samba Share Access Controls

This section deals with how to configure Samba per share access control restrictions. By default samba sets no restrictions on the share itself. Restrictions on the share itself can be set on MS Windows NT4/200x/XP shares. This can be a very effective way to limit who can connect to a share. In the absence of specific restrictions the default setting is to allow the global user *Everyone* Full Control (ie: Full control, Change and Read).

At this time Samba does NOT provide a tool for configuring access control setting on the Share itself. Samba does have the capacity to store and act on access control settings, but the only way to create those settings is to use either the NT4 Server Manager or the Windows 200x MMC for Computer Management.

Samba stores the per share access control settings in a file called `share_info.tdb`. The location of this file on your system will depend on how samba was compiled. The default location for samba's tdb files is under `/usr/local/samba/var`. If the `tdbdump` utility has been compiled and installed on your system then you can examine the contents of this file by: **`tdbdump share_info.tdb`**.

15.1.1. Share Permissions Management

The best tool for the task is platform dependant. Choose the best tool for your environmemt.

15.1.1.1. Windows NT4 Workstation/Server

The tool you need to use to manage share permissions on a Samba server is the NT Server Manager. Server Manager is shipped with Windows NT4 Server products but not with Windows NT4 Workstation. You can obtain the NT Server Manager for MS Windows NT4 Workstation from Microsoft - see details below.

Instructions

1. Launch the NT4 Server Manager, click on the Samba server you want to administer, then from the menu select Computer, then click on the Shared Directories entry.
2. Now click on the share that you wish to manage, then click on the Properties tab, next click on the Permissions tab. Now you can Add or change access control settings as you wish.

15.1.1.2. Windows 200x/XP

On MS Windows NT4/200x/XP system access control lists on the share itself are set using native tools, usually from filemanager. For example, in Windows 200x: right click on the shared folder, then select 'Sharing', then click on 'Permissions'. The default Windows NT4/200x permission allows *Everyone* Full Control on the Share.

MS Windows 200x and later all comes with a tool called the 'Computer Management' snap-in for the Microsoft Management Console (MMC). This tool is located by clicking on `Control Panel -> Administrative Tools -> Computer Management`.

Instructions

1. After launching the MMC with the Computer Management snap-in, click on the menu item 'Action', select 'Connect to another computer'. If you are not logged onto a domain you will be prompted to enter a domain login user identifier and a password. This will authenticate you to the domain. If you where already logged in with administrative privilidge this step is not offered.
2. If the Samba server is not shown in the Select Computer box, then type in the name of the target Samba server in the field 'Name:'. Now click on the [+] next to 'System Tools', then on the [+] next to 'Shared Folders' in the left panel.
3. Now in the right panel, double-click on the share you wish to set access control permissions on. Then click on the tab 'Share Permissions'. It is now possible to add access control entities to the shared folder. Do NOT forget to set what type of access (full control, change, read) you wish to assign for each entry.

Warning

Be careful. If you take away all permissions from the Everyone user without removing this user then effectively no user will be able to access the share. This is a result of what is known as ACL precidence. ie: Everyone with NO ACCESS means that MaryK who is part of the group Everyone will have no access even if this user is given explicit full control access.

15.2. Remote Server Administration

How do I get 'User Manager' and 'Server Manager'?

Since I don't need to buy an NT4 Server, how do I get the 'User Manager for Domains', the 'Server Manager'?

Microsoft distributes a version of these tools called nexus for installation on Windows 9x / Me systems. The tools set includes:

- Server Manager
- User Manager for Domains
- Event Viewer

Click here to download the archived file ftp://ftp.microsoft.com/Softlib/MSLFILES/NEXUS.EXE

The Windows NT 4.0 version of the 'User Manager for Domains' and 'Server Manager' are available from Microsoft via ftp from ftp://ftp.microsoft.com/Softlib/MSLFILES/SRVTOOLS.EXE

15.3. Network Logon Script Magic

This section needs work. Volunteer contributions most welcome. Please send your patches or updates to John Terpstra (mailto:jht@samba.org).

Chapter 16. System and Account Policies

16.1. Creating and Managing System Policies

Under MS Windows platforms, particularly those following the release of MS Windows NT4 and MS Windows 95) it is possible to create a type of file that would be placed in the NETLOGON share of a domain controller. As the client logs onto the network this file is read and the contents initiate changes to the registry of the client machine. This file allows changes to be made to those parts of the registry that affect users, groups of users, or machines.

For MS Windows 9x/Me this file must be called `Config.POL` and may be generated using a tool called `poledit.exe`, better known as the Policy Editor. The policy editor was provided on the Windows 98 installation CD, but dissappeared again with the introduction of MS Windows Me (Millenium Edition). From comments from MS Windows network administrators it would appear that this tool became a part of the MS Windows Me Resource Kit.

MS Windows NT4 Server products include the *System Policy Editor* under the `Start -> Programs -> Administrative Tools` menu item. For MS Windows NT4 and later clients this file must be called `NTConfig.POL`.

New with the introduction of MS Windows 2000 was the Microsoft Management Console or MMC. This tool is the new wave in the ever changing landscape of Microsoft methods for management of network access and security. Every new Microsoft product or technology seems to obsolete the old rules and to introduce newer and more complex tools and methods. To Microsoft's credit though, the MMC does appear to be a step forward, but improved functionality comes at a great price.

Before embarking on the configuration of network and system policies it is highly advisable to read the documentation available from Microsoft's web site regarding Implementing Profiles and Policies in Windows NT 4.0 from http://www.microsoft.com/ntserver/management/deployment/planguide/prof_policies.asp (http://www.microsoft.com/ntserver/management/deployment/planguide/prof_policies.asp) available from Microsoft. There are a large number of documents in addition to this old one that should also be read and understood. Try searching on the Microsoft web site for "Group Policies".

What follows is a very discussion with some helpful notes. The information provided here is incomplete - you are warned.

16.1.1. Windows 9x/Me Policies

You need the Win98 Group Policy Editor to set Group Profiles up under Windows 9x/Me. It can be found on the Original full product Win98 installation CD under `tools/reskit/netadmin/poledit`. Install this using the Add/Remove Programs facility and then click on the 'Have Disk' tab.

Use the Group Policy Editor to create a policy file that specifies the location of user profiles and/or the `My Documents` etc. stuff. Then save these settings in a file called `Config.POL` that needs to be placed in the root of the [NETLOGON] share. If Win98 is configured to log onto the Samba Domain, it will automatically read this file and update the Win9x/Me registry of the machine as it logs on.

Further details are covered in the Win98 Resource Kit documentation.

If you do not take the right steps, then every so often Win9x/Me will check the integrity of the registry and will restore it's settings from the back-up copy of the registry it stores on each Win9x/Me machine. Hence, you will occasionally notice things changing back to the original settings.

Install the group policy handler for Win9x to pick up group policies. Look on the Win98 CD in `\tools\reskit\netadmin\poledit`. Install group policies on a Win9x client by double-clicking `grouppol.inf`. Log off and on again a couple of times and see if Win98 picks up group policies. Unfortunately this needs to be done on every Win9x/Me machine that uses group policies.

16.1.2. Windows NT4 Style Policy Files

To create or edit `ntconfig.pol` you must use the NT Server Policy Editor, **poledit.exe** which is included with NT4 Server but *not NT Workstation*. There is a Policy Editor on a NT4 Workstation but it is not suitable for creating *Domain Policies*. Further, although the Windows 95 Policy Editor can be installed on an NT4 Workstation/Server, it will not work with NT clients. However, the files from the NT Server will run happily enough on an NT4 Workstation.

You need `poledit.exe, common.adm` and `winnt.adm`. It is convenient to put the two *.adm files in the `c:\winnt\inf` directory which is where the binary will look for them unless told otherwise. Note also that that directory is normally 'hidden'.

The Windows NT policy editor is also included with the Service Pack 3 (and later) for Windows NT 4.0. Extract the files using **servicepackname /x**, i.e. that's **Nt4sp6ai.exe /x** for service pack 6a. The policy editor, **poledit.exe** and the associated template files (*.adm) should be extracted as well. It is also possible to downloaded the policy template files for Office97 and get a copy of the policy editor. Another possible location is with the Zero Administration Kit available for download from Microsoft.

16.1.2.1. Registry Tattoos

With NT4 style registry based policy changes, a large number of settings are not automatically reversed as the user logs off. Since the settings that were in the NTConfig.POL file were applied to the client machine registry and that apply to the hive key HKEY_LOCAL_MACHINE are permanent until explicitly reversed. This is known as tattooing. It can have serious consequences down-stream and the administrator must be extremely careful not to lock out the ability to manage the machine at a later date.

16.1.3. MS Windows 200x / XP Professional Policies

Windows NT4 System policies allows setting of registry parameters specific to users, groups and computers (client workstations) that are members of the NT4 style domain. Such policy file will work with MS Windows

2000 / XP clients also.

New to MS Windows 2000 Microsoft introduced a new style of group policy that confers a superset of capabilities compared with NT4 style policies. Obviously, the tool used to create them is different, and the mechanism for implementing them is much changed.

The older NT4 style registry based policies are known as *Administrative Templates* in MS Windows 2000/XP Group Policy Objects (GPOs). The later includes ability to set various security configurations, enforce Internet Explorer browser settings, change and redirect aspects of the users' desktop (including: the location of *My Documents* files (directory), as well as intrinsics of where menu items will appear in the Start menu). An additional new feature is the ability to make available particular software Windows applications to particular users and/or groups.

Remember: NT4 policy files are named `NTConfig.POL` and are stored in the root of the NETLOGON share on the domain controllers. A Windows NT4 user enters a username, a password and selects the domain name to which the logon will attempt to take place. During the logon process the client machine reads the NTConfig.POL file from the NETLOGON share on the authenticating server, modifies the local registry values according to the settings in this file.

Windows 2K GPOs are very feature rich. They are NOT stored in the NETLOGON share, rather part of a Windows 200x policy file is stored in the Active Directory itself and the other part is stored in a shared (and replicated) volume called the SYSVOL folder. This folder is present on all Active Directory domain controllers. The part that is stored in the Active Directory itself is called the group policy container (GPC), and the part that is stored in the replicated share called SYSVOL is known as the group policy template (GPT).

With NT4 clients the policy file is read and executed upon only aas each user log onto the network. MS Windows 200x policies are much more complex - GPOs are processed and applied at client machine startup (machine specific part) and when the user logs onto the network the user specific part is applied. In MS Windows 200x style policy management each machine and/or user may be subject to any number of concurently applicable (and applied) policy sets (GPOs). Active Directory allows the administrator to also set filters over the policy settings. No such equivalent capability exists with NT4 style policy files.

16.1.3.1. Administration of Win2K / XP Policies

Instructions

Instead of using the tool called "The System Policy Editor", commonly called Poledit (from the executable name poledit.exe), GPOs are created and managed using a Microsoft Management Console (MMC) snap-in as follows:

1. Go to the Windows 200x / XP menu `Start->Programs->Administrative Tools` and select the MMC snap-in called "Active Directory Users and Computers"
2. Select the domain or organizational unit (OU) that you wish to manage, then right click to open the context menu for that object, select the properties item.
3. Now left click on the Group Policy tab, then left click on the New tab. Type a name for the new policy you will create.

4. Now left click on the Edit tab to commence the steps needed to create the GPO.

All policy configuration options are controlled through the use of policy administrative templates. These files have a .adm extension, both in NT4 as well as in Windows 200x / XP. Beware however, since the .adm files are NOT interchangible across NT4 and Windows 200x. The later introduces many new features as well as extended definition capabilities. It is well beyond the scope of this documentation to explain how to program .adm files, for that the adminsitrator is referred to the Microsoft Windows Resource Kit for your particular version of MS Windows.

> **Note:** The MS Windows 2000 Resource Kit contains a tool called gpolmig.exe. This tool can be used to migrate an NT4 NTConfig.POL file into a Windows 200x style GPO. Be VERY careful how you use this powerful tool. Please refer to the resource kit manuals for specific usage information.

16.2. Managing Account/User Policies

Policies can define a specific user's settings or the settings for a group of users. The resulting policy file contains the registry settings for all users, groups, and computers that will be using the policy file. Separate policy files for each user, group, or computer are not not necessary.

If you create a policy that will be automatically downloaded from validating domain controllers, you should name the file NTconfig.POL. As system administrator, you have the option of renaming the policy file and, by modifying the Windows NT-based workstation, directing the computer to update the policy from a manual path. You can do this by either manually changing the registry or by using the System Policy Editor. This path can even be a local path such that each machine has its own policy file, but if a change is necessary to all machines, this change must be made individually to each workstation.

When a Windows NT4/200x/XP machine logs onto the network the NETLOGON share on the authenticating domain controller for the presence of the NTConfig.POL file. If one exists it is downloaded, parsed and then applied to the user's part of the registry.

MS Windows 200x/XP clients that log onto an MS Windows Active Directory security domain may additionally, acquire policy settings through Group Policy Objects (GPOs) that are defined and stored in Active Directory itself. The key benefit of using AS GPOs is that they impose no registry *tatooing* effect. This has considerable advanage compared with the use of NTConfig.POL (NT4) style policy updates.

Inaddition to user access controls that may be imposed or applied via system and/or group policies in a manner that works in conjunction with user profiles, the user management environment under MS Windows NT4/200x/XP allows per domain as well as per user account restrictions to be applied. Common restrictions that are frequently used includes:

Logon Hours
Password Aging
Permitted Logon from certain machines only

Account type (Local or Global)
User Rights

16.2.1. With Windows NT4/200x

The tools that may be used to configure these types of controls from the MS Windows environment are: The NT4 User Manager for domains, the NT4 System and Group Policy Editor, the registry editor (regedt32.exe). Under MS Windows 200x/XP this is done using the Microsoft Managment Console (MMC) with approapriate "snap-ins", the registry editor, and potentially also the NT4 System and Group Policy Editor.

16.2.2. With a Samba PDC

With a Samba Domain Controller, the new tools for managing of user account and policy information includes: `smbpasswd, pdbedit, smbgroupedit, net, rpcclient.`. The administrator should read the man pages for these tools and become familiar with their use.

Chapter 17. Desktop Profile Management

17.1. Roaming Profiles

Warning

Roaming profiles support is different for Win9x / Me and Windows NT4/200x.

Before discussing how to configure roaming profiles, it is useful to see how Windows 9x / Me and Windows NT4/200x clients implement these features.

Windows 9x / Me clients send a NetUserGetInfo request to the server to get the user's profiles location. However, the response does not have room for a separate profiles location field, only the user's home share. This means that Win9X/Me profiles are restricted to being stored in the user's home directory.

Windows NT4/200x clients send a NetSAMLogon RPC request, which contains many fields, including a separate field for the location of the user's profiles.

17.1.1. Samba Configuration for Profile Handling

This section documents how to configure Samba for MS Windows client profile support.

17.1.1.1. NT4/200x User Profiles

To support Windowns NT4/200x clients, in the [global] section of smb.conf set the following (for example):

```
logon path = \\profileserver\profileshare\profilepath\%U\moreprofilepath
```

This is typically implemented like:

```
logon path = \\%L\Profiles\%u
```

where %L translates to the name of the Samba server and %u translates to the user name

The default for this option is \\%N\%U\profile, namely \\sambaserver\username\profile. The \\N%\%U service is created automatically by the [homes] service. If you are using a samba server for the profiles, you _must_ make the share specified in the logon path browseable. Please refer to the man page for smb.conf in respect of the different symantics of %L and %N, as well as %U and %u.

> **Note:** MS Windows NT/2K clients at times do not disconnect a connection to a server between logons. It is recommended to NOT use the **homes** meta-service name as part of the profile share path.

17.1.1.2. Windows 9x / Me User Profiles

To support Windows 9x / Me clients, you must use the "logon home" parameter. Samba has now been fixed so that **`net use /home`** now works as well, and it, too, relies on the **logon home<** parameter.

By using the logon home parameter, you are restricted to putting Win9x / Me profiles in the user's home directory. But wait! There is a trick you can use. If you set the following in the **[global]** section of your `smb.conf` file:

```
logon home = \\%L\%U\.profiles
```

then your Windows 9x / Me clients will dutifully put their clients in a subdirectory of your home directory called `.profiles` (thus making them hidden).

Not only that, but **`net use/home`** will also work, because of a feature in Windows 9x / Me. It removes any directory stuff off the end of the home directory area and only uses the server and share portion. That is, it looks like you specified \\%L\%U for **logon home**.

17.1.1.3. Mixed Windows 9x / Me and Windows NT4/200x User Profiles

You can support profiles for both Win9X and WinNT clients by setting both the **logon home** and **logon path** parameters. For example:

```
logon home = \\%L\%u\.profiles
logon path = \\%L\profiles\%u
```

17.1.2. Windows Client Profile Configuration Information

17.1.2.1. Windows 9x / Me Profile Setup

When a user first logs in on Windows 9X, the file user.DAT is created, as are folders "Start Menu", "Desktop", "Programs" and "Nethood". These directories and their contents will be merged with the local versions stored in c:\windows\profiles\username on subsequent logins, taking the most recent from each. You will need to use the [global] options "preserve case = yes", "short preserve case = yes" and "case sensitive = no" in order to maintain capital letters in shortcuts in any of the profile folders.

The user.DAT file contains all the user's preferences. If you wish to enforce a set of preferences, rename their user.DAT file to user.MAN, and deny them write access to this file.

1. On the Windows 9x / Me machine, go to Control Panel -> Passwords and select the User Profiles tab. Select the required level of roaming preferences. Press OK, but do _not_ allow the computer to reboot.
2. On the Windows 9x / Me machine, go to Control Panel -> Network -> Client for Microsoft Networks -> Preferences. Select ’Log on to NT Domain’. Then, ensure that the Primary Logon is ’Client for Microsoft Networks’. Press OK, and this time allow the computer to reboot.

Under Windows 9x / Me Profiles are downloaded from the Primary Logon. If you have the Primary Logon as ’Client for Novell Networks’, then the profiles and logon script will be downloaded from your Novell Server. If you have the Primary Logon as ’Windows Logon’, then the profiles will be loaded from the local machine - a bit against the concept of roaming profiles, it would seem!

You will now find that the Microsoft Networks Login box contains [user, password, domain] instead of just [user, password]. Type in the samba server’s domain name (or any other domain known to exist, but bear in mind that the user will be authenticated against this domain and profiles downloaded from it, if that domain logon server supports it), user name and user’s password.

Once the user has been successfully validated, the Windows 9x / Me machine will inform you that ’The user has not logged on before’ and asks you if you wish to save the user’s preferences? Select ’yes’.

Once the Windows 9x / Me client comes up with the desktop, you should be able to examine the contents of the directory specified in the "logon path" on the samba server and verify that the "Desktop", "Start Menu", "Programs" and "Nethood" folders have been created.

These folders will be cached locally on the client, and updated when the user logs off (if you haven’t made them read-only by then). You will find that if the user creates further folders or short-cuts, that the client will merge the profile contents downloaded with the contents of the profile directory already on the local client, taking the newest folders and short-cuts from each set.

If you have made the folders / files read-only on the samba server, then you will get errors from the Windows 9x / Me machine on logon and logout, as it attempts to merge the local and the remote profile. Basically, if you have any errors reported by the Windows 9x / Me machine, check the Unix file permissions and ownership rights on the profile directory contents, on the samba server.

If you have problems creating user profiles, you can reset the user’s local desktop cache, as shown below. When this user then next logs in, they will be told that they are logging in "for the first time".

1. instead of logging in under the [user, password, domain] dialog, press escape.
2. run the regedit.exe program, and look in:

   HKEY_LOCAL_MACHINE\Windows\CurrentVersion\ProfileList

   you will find an entry, for each user, of ProfilePath. Note the contents of this key (likely to be c:\windows\profiles\username), then delete the key ProfilePath for the required user. [Exit the registry editor].

3. *WARNING* - before deleting the contents of the directory listed in the ProfilePath (this is likely to be `c:\windows\profiles\username)`, ask them if they have any important files stored on their desktop or in their start menu. Delete the contents of the directory ProfilePath (making a backup if any of the files are needed).

   This will have the effect of removing the local (read-only hidden system file) user.DAT in their profile directory, as well as the local "desktop", "nethood", "start menu" and "programs" folders.

4. search for the user's .PWL password-caching file in the c:\windows directory, and delete it.
5. log off the windows 9x / Me client.
6. check the contents of the profile path (see "logon path" described above), and delete the user.DAT or user.MAN file for the user, making a backup if required.

If all else fails, increase samba's debug log levels to between 3 and 10, and / or run a packet trace program such as ethereal or netmon.exe, and look for error messages.

If you have access to an Windows NT4/200x server, then first set up roaming profiles and / or netlogons on the Windows NT4/200x server. Make a packet trace, or examine the example packet traces provided with Windows NT4/200x server, and see what the differences are with the equivalent samba trace.

17.1.2.2. Windows NT4 Workstation

When a user first logs in to a Windows NT Workstation, the profile NTuser.DAT is created. The profile location can be now specified through the "logon path" parameter.

There is a parameter that is now available for use with NT Profiles: "logon drive". This should be set to `H:` or any other drive, and should be used in conjunction with the new "logon home" parameter.

The entry for the NT4 profile is a _directory_ not a file. The NT help on profiles mentions that a directory is also created with a .PDS extension. The user, while logging in, must have write permission to create the full profile path (and the folder with the .PDS extension for those situations where it might be created.)

In the profile directory, Windows NT4 creates more folders than Windows 9x / Me. It creates "Application Data" and others, as well as "Desktop", "Nethood", "Start Menu" and "Programs". The profile itself is stored in a file NTuser.DAT. Nothing appears to be stored in the .PDS directory, and its purpose is currently unknown.

You can use the System Control Panel to copy a local profile onto a samba server (see NT Help on profiles: it is also capable of firing up the correct location in the System Control Panel for you). The NT Help file also mentions that renaming NTuser.DAT to NTuser.MAN turns a profile into a mandatory one.

The case of the profile is significant. The file must be called NTuser.DAT or, for a mandatory profile, NTuser.MAN.

17.1.2.3. Windows 2000/XP Professional

You must first convert the profile from a local profile to a domain profile on the MS Windows workstation as follows:

- Log on as the LOCAL workstation administrator.
- Right click on the ’My Computer’ Icon, select ’Properties’
- Click on the ’User Profiles’ tab
- Select the profile you wish to convert (click on it once)
- Click on the button ’Copy To’
- In the "Permitted to use" box, click on the ’Change’ button.
- Click on the ’Look in" area that lists the machine name, when you click here it will open up a selection box. Click on the domain to which the profile must be accessible.

  > **Note:** You will need to log on if a logon box opens up. Eg: In the connect as: MIDEARTH\root, password: mypassword.

- To make the profile capable of being used by anyone select ’Everyone’
- Click OK. The Selection box will close.
- Now click on the ’Ok’ button to create the profile in the path you nominated.

Done. You now have a profile that can be editted using the samba-3.0.0 `profiles` tool.

> **Note:** Under NT/2K the use of mandotory profiles forces the use of MS Exchange storage of mail data. That keeps desktop profiles usable.

- This is a security check new to Windows XP (or maybe only Windows XP service pack 1). It can be disabled via a group policy in Active Directory. The policy is:

  "Computer Configuration\Administrative Templates\System\User Profiles\Do not check for user ownership of Roaming Profile Folders"

  ...and it should be set to "Enabled". Does the new version of samba have an Active Directory analogue? If so, then you may be able to set the policy through this.

  If you cannot set group policies in samba, then you may be able to set the policy locally on each machine. If you want to try this, then do the following (N.B. I don’t know for sure that this will work in the same way as a domain group policy):

- On the XP workstation log in with an Administrator account.
- Click: "Start", "Run"

- Type: "mmc"
- Click: "OK"
- A Microsoft Management Console should appear.
- Click: File, "Add/Remove Snap-in...", "Add"
- Double-Click: "Group Policy"
- Click: "Finish", "Close"
- Click: "OK"
- In the "Console Root" window:
- Expand: "Local Computer Policy", "Computer Configuration",
- "Administrative Templates", "System", "User Profiles"
- Double-Click: "Do not check for user ownership of Roaming Profile
- Folders"
- Select: "Enabled"
- Click: OK"
- Close the whole console. You do not need to save the settings (this refers to the console settings rather than the policies you have changed).
- Reboot

17.1.3. Sharing Profiles between W9x/Me and NT4/200x/XP workstations

Sharing of desktop profiles between Windows versions is NOT recommended. Desktop profiles are an evolving phenomenon and profiles for later versions of MS Windows clients add features that may interfere with earlier versions of MS Windows clients. Probably the more salient reason to NOT mix profiles is that when logging off an earlier version of MS Windows the older format of profile contents may overwrite information that belongs to the newer version resulting in loss of profile information content when that user logs on again with the newer version of MS Windows.

If you then want to share the same Start Menu / Desktop with W9x/Me, you will need to specify a common location for the profiles. The smb.conf parameters that need to be common are *logon path* and *logon home*.

If you have this set up correctly, you will find separate user.DAT and NTuser.DAT files in the same profile directory.

17.1.4. Profile Migration from Windows NT4/200x Server to Samba

There is nothing to stop you specifying any path that you like for the location of users’ profiles. Therefore, you could specify that the profile be stored on a samba server, or any other SMB server, as long as that SMB server supports encrypted passwords.

17.1.4.1. Windows NT4 Profile Management Tools

Unfortunately, the Resource Kit information is specific to the version of MS Windows NT4/200x. The correct resource kit is required for each platform.

Here is a quick guide:

- On your NT4 Domain Controller, right click on 'My Computer', then select the tab labelled 'User Profiles'.
- Select a user profile you want to migrate and click on it.

> **Note:** I am using the term "migrate" lossely. You can copy a profile to create a group profile. You can give the user 'Everyone' rights to the profile you copy this to. That is what you need to do, since your samba domain is not a member of a trust relationship with your NT4 PDC.

- Click the 'Copy To' button.
- In the box labelled 'Copy Profile to' add your new path, eg: `c:\temp\foobar`
- Click on the button labelled 'Change' in the "Permitted to use" box.
- Click on the group 'Everyone' and then click OK. This closes the 'chose user' box.
- Now click OK.

Follow the above for every profile you need to migrate.

17.1.4.2. Side bar Notes

You should obtain the SID of your NT4 domain. You can use smbpasswd to do this. Read the man page.

With Samba-3.0.0 alpha code you can import all you NT4 domain accounts using the net samsync method. This way you can retain your profile settings as well as all your users.

17.1.4.3. moveuser.exe

The W2K professional resource kit has moveuser.exe. moveuser.exe changes the security of a profile from one user to another. This allows the account domain to change, and/or the user name to change.

17.1.4.4. Get SID

You can identify the SID by using GetSID.exe from the Windows NT Server 4.0 Resource Kit.

Windows NT 4.0 stores the local profile information in the registry under the following key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList

Under the ProfileList key, there will be subkeys named with the SIDs of the users who have logged on to this computer. (To find the profile information for the user whose locally cached profile you want to move, find the SID for the user with the GetSID.exe utility.) Inside of the appropriate user's subkey, you will see a string value named ProfileImagePath.

17.2. Mandatory profiles

A Mandatory Profile is a profile that the user does NOT have the ability to overwrite. During the user's session it may be possible to change the desktop environment, but as the user logs out all changes made will be lost. If it is desired to NOT allow the user any ability to change the desktop environment then this must be done through policy settings. See previous chapter.

> **Note:** Under NO circumstances should the profile directory (or it's contents) be made read-only as this may render the profile un-usable.

For MS Windows NT4/200x/XP the above method can be used to create mandatory profiles also. To convert a group profile into a mandatory profile simply locate the NTUser.DAT file in the copied profile and rename it to NTUser.MAN.

For MS Windows 9x / Me it is the User.DAT file that must be renamed to User.MAN to affect a mandatory profile.

17.3. Creating/Managing Group Profiles

Most organisations are arranged into departments. There is a nice benenfit in this fact since usually most users in a department will require the same desktop applications and the same desktop layout. MS Windows NT4/200x/XP will allow the use of Group Profiles. A Group Profile is a profile that is created firstly using a template (example) user. Then using the profile migration tool (see above) the profile is assigned access rights for the user group that needs to be given access to the group profile.

The next step is rather important. PLEASE NOTE: Instead of assigning a group profile to users (ie: Using User Manager) on a "per user" basis, the group itself is assigned the now modified profile.

> **Note:** Be careful with group profiles, if the user who is a member of a group also has a personal profile, then the result will be a fusion (merge) of the two.

17.4. Default Profile for Windows Users

MS Windows 9x / Me and NT4/200x/XP will use a default profile for any user for whom a profile does not already exist. Armed with a knowledge of where the default profile is located on the Windows workstation, and knowing which registry keys affect the path from which the default profile is created, it is possible to modify the default profile to one that has been optimised for the site. This has significant administrative advantages.

17.4.1. MS Windows 9x/Me

To enable default per use profiles in Windows 9x / Me you can either use the Windows 98 System Policy Editor or change the registry directly.

To enable default per user profiles in Windows 9x / Me, launch the System Policy Editor, then select File -> Open Registry, then click on the Local Computer icon, click on Windows 98 System, select User Profiles, click on the enable box. Do not forget to save the registry changes.

To modify the registry directly, launch the Registry Editor (regedit.exe), select the hive `HKEY_LOCAL_MACHINE\Network\Logon`. Now add a DWORD type key with the name "User Profiles", to enable user profiles set the value to 1, to disable user profiles set it to 0.

17.4.1.1. How User Profiles Are Handled in Windows 9x / Me?

When a user logs on to a Windows 9x / Me machine, the local profile path, `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\ProfileList`, is checked for an existing entry for that user:

If the user has an entry in this registry location, Windows 9x / Me checks for a locally cached version of the user profile. Windows 9x / Me also checks the user's home directory (or other specified directory if the location has been modified) on the server for the User Profile. If a profile exists in both locations, the newer of the two is used. If the User Profile exists on the server, but does not exist on the local machine, the profile on the server is downloaded and used. If the User Profile only exists on the local machine, that copy is used.

If a User Profile is not found in either location, the Default User Profile from the Windows 9x / Me machine is used and is copied to a newly created folder for the logged on user. At log off, any changes that the user made are written to the user's local profile. If the user has a roaming profile, the changes are written to the user's profile on the server.

17.4.2. MS Windows NT4 Workstation

On MS Windows NT4 the default user profile is obtained from the location `%SystemRoot%\Profiles` which in a default installation will translate to `C:\WinNT\Profiles`. Under this directory on a clean install there will be three (3) directories: `Administrator, All Users, Default User`.

The `All Users` directory contains menu settings that are common across all system users. The `Default User` directory contains menu entries that are customisable per user depending on the profile settings chosen/created.

When a new user first logs onto an MS Windows NT4 machine a new profile is created from:

All Users settings
Default User settings (contains the default NTUser.DAT file)

When a user logs onto an MS Windows NT4 machine that is a member of a Microsoft security domain the following steps are followed in respect of profile handling:

1. The users' account information which is obtained during the logon process contains the location of the users' desktop profile. The profile path may be local to the machine or it may be located on a network share. If there exists a profile at the location of the path from the user account, then this profile is copied to the location `%SystemRoot%\Profiles\%USERNAME%`. This profile then inherits the settings in the `All Users` profile in the `%SystemRoot%\Profiles` location.
2. If the user account has a profile path, but at it's location a profile does not exist, then a new profile is created in the `%SystemRoot%\Profiles\%USERNAME%` directory from reading the `Default User` profile.
3. If the NETLOGON share on the authenticating server (logon server) contains a policy file (`NTConfig.POL`) then it's contents are applied to the `NTUser.DAT` which is applied to the `HKEY_CURRENT_USER` part of the registry.
4. When the user logs out, if the profile is set to be a roaming profile it will be written out to the location of the profile. The `NTuser.DAT` file is then re-created from the contents of the `HKEY_CURRENT_USER` contents. Thus, should there not exist in the NETLOGON share an `NTConfig.POL` at the next logon, the effect of the provious `NTConfig.POL` will still be held in the profile. The effect of this is known as *tatooing*.

MS Windows NT4 profiles may be *Local* or *Roaming*. A Local profile will stored in the `%SystemRoot%\Profiles\%USERNAME%` location. A roaming profile will also remain stored in the same way, unless the following registry key is created:

```
HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\Windows NT\CurrentVersion\winlogon\
"DeleteRoamingCache"=dword:00000001
```

In which case, the local copy (in `%SystemRoot%\Profiles\%USERNAME%`) will be deleted on logout.

Under MS Windows NT4 default locations for common resources (like `My Documents` may be redirected to a network share by modifying the following registry keys. These changes may be affected via use of the System Policy Editor (to do so may require that you create your owns template extension for the policy editor to allow this to be done through the GUI. Another way to do this is by way of first creating a default user profile, then while logged in as that user, run regedt32 to edit the key settings.

The Registry Hive key that affects the behaviour of folders that are part of the default user profile are controlled by entries on Windows NT4 is:

```
        HKEY_CURRENT_USER
                \Software
```

```
                        \Microsoft
                                \Windows
                                        \CurrentVersion
                                                \Explorer
                                                        \User Shell Folders\
```

The above hive key contains a list of automatically managed folders. The default entries are:

```
        Name            Default Value
        --------------  -----------------------------------------
        AppData         %USERPROFILE%\Application Data
        Desktop         %USERPROFILE%\Desktop
        Favorites       %USERPROFILE%\Favorites
        NetHood         %USERPROFILE%\NetHood
        PrintHood       %USERPROFILE%\PrintHood
        Programs        %USERPROFILE%\Start Menu\Programs
        Recent          %USERPROFILE%\Recent
        SendTo          %USERPROFILE%\SendTo
        Start Menu      %USERPROFILE%\Start Menu
        Startup         %USERPROFILE%\Start Menu\Programs\Startup
```

The registry key that contains the location of the default profile settings is:

```
HKEY_LOCAL_MACHINE
\SOFTWARE
\Microsoft
\Windows
\CurrentVersion
\Explorer
\User Shell Folders
```

The default entries are:

```
Common Desktop %SystemRoot%\Profiles\All Users\Desktop
Common Programs %SystemRoot%\Profiles\All Users\Programs
Common Start Menu %SystemRoot%\Profiles\All Users\Start Menu
Common Startu p %SystemRoot%\Profiles\All Users\Start Menu\Progams\Startup
```

17.4.3. MS Windows 200x/XP

> **Note:** MS Windows XP Home Edition does use default per user profiles, but can not participate in domain security, can not log onto an NT/ADS style domain, and thus can obtain the profile only from itself. While there are benefits in doing this the beauty of those MS Windows clients that CAN participate in domain logon processes allows the administrator to create a global default profile and to enforce it through the use of Group Policy Objects (GPOs).

When a new user first logs onto MS Windows 200x/XP machine the default profile is obtained from `C:\Documents and Settings\Default User`. The administrator can modify (or change the contents of this location and MS Windows 200x/XP will gladly user it. This is far from the optimum arrangement since it will involve copying a new default profile to every MS Windows 200x/XP client workstation.

When MS Windows 200x/XP participate in a domain security context, and if the default user profile is not found, then the client will search for a default profile in the NETLOGON share of the authenticating server. ie: In MS Windows parlance: `%LOGONSERVER%\NETLOGON\Default User` and if one exits there it will copy this to the workstation to the `C:\Documents and Settings\` under the Windows login name of the user.

> **Note:** This path translates, in Samba parlance, to the smb.conf [NETLOGON] share. The directory should be created at the root of this share and msut be called `Default Profile`.

If a default profile does not exist in this location then MS Windows 200x/XP will use the local default profile.

On loging out, the users' desktop profile will be stored to the location specified in the registry settings that pertain to the user. If no specific policies have been created, or passed to the client during the login process (as Samba does automatically), then the user's profile will be written to the local machine only under the path `C:\Documents and Settings\%USERNAME%`.

Those wishing to modify the default behaviour can do so through up to three methods:

- Modify the registry keys on the local machine manually and place the new default profile in the NETLOGON share root - NOT recommended as it is maintenance intensive.
- Create an NT4 style NTConfig.POL file that specified this behaviour and locate this file in the root of the NETLOGON share along with the new default profile.
- Create a GPO that enforces this through Active Directory, and place the new default profile in the NETLOGON share.

The Registry Hive key that affects the behaviour of folders that are part of the default user profile are controlled by entries on Windows 200x/XP is:

```
HKEY_CURRENT_USER
\Software
\Microsoft
\Windows
\CurrentVersion
\Explorer
\User Shell Folders\
```

The above hive key contains a list of automatically managed folders. The default entries are:

```
Name Default Value
-------------- -----------------------------------------
AppData %USERPROFILE%\Application Data
Cache %USERPROFILE%\Local Settings\Temporary Internet Files
Cookies %USERPROFILE%\Cookies
Desktop %USERPROFILE%\Desktop
Favorites %USERPROFILE%\Favorites
History %USERPROFILE%\Local Settings\History
Local AppData %USERPROFILE%\Local Settings\Application Data
Local Settings %USERPROFILE%\Local Settings
My Pictures %USERPROFILE%\My Documents\My Pictures
NetHood %USERPROFILE%\NetHood
Personal %USERPROFILE%\My Documents
PrintHood %USERPROFILE%\PrintHood
Programs %USERPROFILE%\Start Menu\Programs
Recent %USERPROFILE%\Recent
SendTo %USERPROFILE%\SendTo
Start Menu %USERPROFILE%\Start Menu
Startup %USERPROFILE%\Start Menu\Programs\Startup
Templates %USERPROFILE%\Templates
```

There is also an entry called "Default" that has no value set. The default entry is of type REG_SZ, all the others are of type REG_EXPAND_SZ.

It makes a huge difference to the speed of handling roaming user profiles if all the folders are stored on a dedicated location on a network server. This means that it will NOT be necessary to write Outlook PST file over the network for every login and logout.

To set this to a network location you could use the following examples:

`%LOGONSERVER%\%USERNAME%\Default Folders`

This would store the folders in the user's home directory under a directory called "Default Folders" You could also use:

`\\SambaServer\FolderShare\%USERNAME%`

in which case the default folders will be stored in the server named *SambaServer* in the share called *FolderShare* under a directory that has the name of the MS Windows user as seen by the Linux/Unix file system.

Please note that once you have created a default profile share, you MUST migrate a user's profile (default or custom) to it.

MS Windows 200x/XP profiles may be *Local* or *Roaming*. A roaming profile will be cached locally unless the following registry key is created:

```
HKEY_LOCAL_MACHINE\SYSTEM\Software\Microsoft\Windows NT\CurrentVersion\winlogon\
"DeleteRoamingCache"=dword:00000001
```

In which case, the local cache copy will be deleted on logout.

Chapter 18. Interdomain Trust Relationships

Samba-3 supports NT4 style domain trust relationships. This is feature that many sites will want to use if they migrate to Samba-3 from and NT4 style domain and do NOT want to adopt Active Directory or an LDAP based authentication back end. This section explains some background information regarding trust relationships and how to create them. It is now possible for Samba3 to NT4 trust (and vica versa), as well as Samba3 to Samba3 trusts.

18.1. Trust Relationship Background

MS Windows NT3.x/4.0 type security domains employ a non-hierchical security structure. The limitations of this architecture as it affects the scalability of MS Windows networking in large organisations is well known. Additionally, the flat-name space that results from this design significantly impacts the delegation of administrative responsibilities in large and diverse organisations.

Microsoft developed Active Directory Service (ADS), based on Kerberos and LDAP, as a means of circumventing the limitations of the older technologies. Not every organisation is ready or willing to embrace ADS. For small companies the older NT4 style domain security paradigm is quite adequate, there thus remains an entrenched user base for whom there is no direct desire to go through a disruptive change to adopt ADS.

Microsoft introduced with MS Windows NT the ability to allow differing security domains to affect a mechanism so that users from one domain may be given access rights and privilidges in another domain. The language that describes this capability is couched in terms of *Trusts*. Specifically, one domain will *trust* the users from another domain. The domain from which users are available to another security domain is said to be a trusted domain. The domain in which those users have assigned rights and privilidges is the trusting domain. With NT3.x/4.0 all trust relationships are always in one direction only, thus if users in both domains are to have privilidges and rights in each others' domain, then it is necessary to establish two (2) relationships, one in each direction.

In an NT4 style MS security domain, all trusts are non-transitive. This means that if there are three (3) domains (let's call them RED, WHITE, and BLUE) where RED and WHITE have a trust relationship, and WHITE and BLUE have a trust relationship, then it holds that there is no implied trust between the RED and BLUE domains. ie: Relationships are explicit and not transitive.

New to MS Windows 2000 ADS security contexts is the fact that trust relationships are two-way by default. Also, all inter-ADS domain trusts are transitive. In the case of the RED, WHITE and BLUE domains above, with Windows 2000 and ADS the RED and BLUE domains CAN trust each other. This is an inherent feature of ADS domains.

18.2. MS Windows NT4 Trust Configuration

There are two steps to creating an inter-domain trust relationship.

18.2.1. NT4 as the Trusting Domain

For MS Windows NT4, all domain trust relationships are configured using the Domain User Manager. To affect a two way trust relationship it is necessary for each domain administrator to make available (for use by an external domain) it’s security resources. This is done from the Domain User Manager Policies entry on the menu bar. From the Policy menu, select Trust Relationships, then next to the lower box that is labelled "Permitted to Trust this Domain" are two buttons, "Add" and "Remove". The "Add" button will open a panel in which needs to be entered the remote domain that will be able to assign user rights to your domain. In addition it is necessary to enter a password that is specific to this trust relationship. The password is added twice.

18.2.2. NT4 as the Trusted Domain

A trust relationship will work only when the other (trusting) domain makes the appropriate connections with the trusted domain. To consumate the trust relationship the administrator will launch the Domain User Manager, from the menu select Policies, then select Trust Relationships, then click on the "Add" button that is next to the box that is labelled "Trusted Domains". A panel will open in which must be entered the name of the remote domain as well as the password assigned to that trust.

18.3. Configuring Samba Domain Trusts

This descitpion is meant to be a fairly short introduction about how to set up a Samba server so that it could participate in interdomain trust relationships. Trust relationship support in Samba is in its early stage, so lot of things don’t work yet. Paricularly, the contents of this document applies to NT4-style trusts.

Each of the procedures described below is treated as they were performed with Windows NT4 Server on one end. The other end could just as well be another Samba3 domain. It can be clearly seen, after reading this document, that combining Samba-specific parts of what’s written below leads to trust between domains in purely Samba environment.

18.3.1. Samba3 as the Trusting Domain

In order to set Samba PDC to be trusted party of the relationship first you need to create special account for domain that will be the trusting party. To do that, you can use ’smbpasswd’ utility. Creating the trusted domain account is very similiar to creating the connection to the trusting machine’s account. Suppose, your domain is called SAMBA, and the remote domain is called RUMBA. Your first step will be to issue this command from your favourite shell:

```
deity# smbpasswd -a -i rumba
New SMB password: XXXXXXXX
Retype SMB password: XXXXXXXX
Added user rumba$
```

```
where:
-a means to add a new account into the passdb database
-i means create this account with the Inter-Domain trust flag

The account name will be 'rumba$' (the name ofthe remote domain)
```

fter issuing this command you'll be asked for typing account's password. You can use any password you want, but be aware that Windows NT will not change this password until 7 days have passed since account creating. After command returns successfully, you can look at your new account's entry (in the way depending on your configuration) and see that account's name is really RUMBA$ and it has 'I' flag in the flags field. Now you're ready to confirm the trust by establishing it from Windows NT Server.

Open 'User Manager for Domains' and from menu 'Policies' select 'Trust Relationships...'. Right beside 'Trusted domains' list press 'Add...' button. You'll be prompted for trusted domain name and the relationship's password. Type in SAMBA, as this is your domain name and the password you've just used during account creation. Press OK and if everything went fine, you will see 'Trusted domain relationship successfully established' message. Well done.

18.3.2. Samba3 as the Trusted Domain

This time activities are somewhat reversed. Again, we'll assume that your domain controlled by Samba PDC is called SAMBA and NT-controlled domain is called RUMBA.

The very first thing is to add account for SAMBA domain on RUMBA's PDC.

Launch the Domain User Manager, then from the menu select 'Policies', 'Trust Relationships'. Now, next to 'Trusted Domains' box press the 'Add' button, and type in the name of the trusted domein (SAMBA) and password securing the relationship.

Password can be arbitrarily chosen the more, because it's easy to change it from Samba server whenever you want. After confirming password your account is ready and waiting. Now it's Samba's turn.

Using your favourite shell while being logged on as root, issue this command:

```
deity#  net rpc trustdom establish rumba
```

You'll be prompted for password you've just typed on your Windows NT4 Server box. Don't worry if you will see the error message with returned code of `NT_STATUS_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT`. It means the password you gave is correct and the NT4 Server says the account is ready for trusting your domain and not for ordinary connection. After that, be patient it can take a while (especially in large networks), you should see 'Success' message. Contgratulations! Your trust relationship has just been established.

Note: Note that you have to run this command as root, since you need write access to your secrets.tdb file.

Chapter 19. PAM Configuration for Centrally Managed Authentication

19.1. Samba and PAM

A number of Unix systems (eg: Sun Solaris), as well as the xxxxBSD family and Linux, now utilize the Pluggable Authentication Modules (PAM) facility to provide all authentication, authorization and resource control services. Prior to the introduction of PAM, a decision to use an alternative to the system password database (`/etc/passwd`) would require the provision of alternatives for all programs that provide security services. Such a choice would involve provision of alternatives to such programs as: **login**, **passwd**, **chown**, etc.

PAM provides a mechanism that disconnects these security programs from the underlying authentication/authorization infrastructure. PAM is configured either through one file `/etc/pam.conf` (Solaris), or by editing individual files that are located in `/etc/pam.d`.

> **Note:** If the PAM authentication module (loadable link library file) is located in the default location then it is not necessary to specify the path. In the case of Linux, the default location is `/lib/security`. If the module is located other than default then the path may be specified as:
>
> ```
> auth       required      /other_path/pam_strange_module.so
> ```

The following is an example `/etc/pam.d/login` configuration file. This example had all options been uncommented is probably not usable as it stacks many conditions before allowing successful completion of the login process. Essentially all conditions can be disabled by commenting them out except the calls to `pam_pwdb.so`.

```
#%PAM-1.0
# The PAM configuration file for the ‘login’ service
#
auth  required pam_securetty.so
auth  required pam_nologin.so
# auth  required pam_dialup.so
# auth  optional pam_mail.so
auth required pam_pwdb.so shadow md5
# account     requisite   pam_time.so
account required pam_pwdb.so
session required pam_pwdb.so
# session  optional pam_lastlog.so
# password    required    pam_cracklib.so retry=3
password required pam_pwdb.so shadow md5
```

PAM allows use of replacable modules. Those available on a sample system include:

```
$/bin/ls /lib/security

pam_access.so    pam_ftp.so          pam_limits.so
pam_ncp_auth.so  pam_rhosts_auth.so  pam_stress.so
pam_cracklib.so  pam_group.so        pam_listfile.so
pam_nologin.so   pam_rootok.so       pam_tally.so
pam_deny.so      pam_issue.so        pam_mail.so
pam_permit.so    pam_securetty.so    pam_time.so
pam_dialup.so    pam_lastlog.so      pam_mkhomedir.so
pam_pwdb.so      pam_shells.so       pam_unix.so
pam_env.so       pam_ldap.so         pam_motd.so
pam_radius.so    pam_smbpass.so      pam_unix_acct.so
pam_wheel.so     pam_unix_auth.so    pam_unix_passwd.so
pam_userdb.so    pam_warn.so         pam_unix_session.so
```

The following example for the login program replaces the use of the `pam_pwdb.so` module which uses the system password database (`/etc/passwd`, `/etc/shadow`, `/etc/group`) with the module `pam_smbpass.so` which uses the Samba database which contains the Microsoft MD4 encrypted password hashes. This database is stored in either `/usr/local/samba/private/smbpasswd`, `/etc/samba/smbpasswd`, or in `/etc/samba.d/smbpasswd`, depending on the Samba implementation for your Unix/Linux system. The `pam_smbpass.so` module is provided by Samba version 2.2.1 or later. It can be compiled by specifying the **--with-pam_smbpass** options when running Samba's `configure` script. For more information on the `pam_smbpass` module, see the documentation in the `source/pam_smbpass` directory of the Samba source distribution.

```
#%PAM-1.0
# The PAM configuration file for the `login' service
#
auth required pam_smbpass.so nodelay
account required pam_smbpass.so nodelay
session required pam_smbpass.so nodelay
password required pam_smbpass.so nodelay
```

The following is the PAM configuration file for a particular Linux system. The default condition uses `pam_pwdb.so`.

```
#%PAM-1.0
# The PAM configuration file for the `samba' service
#
auth       required     pam_pwdb.so nullok nodelay shadow audit
account    required     pam_pwdb.so audit nodelay
session    required     pam_pwdb.so nodelay
password   required     pam_pwdb.so shadow md5
```

In the following example the decision has been made to use the smbpasswd database even for basic samba authentication. Such a decision could also be made for the passwd program and would thus allow the smbpasswd passwords to be changed using the passwd program.

```
#%PAM-1.0
# The PAM configuration file for the ‘samba’ service
#
auth       required     pam_smbpass.so nodelay
account    required     pam_pwdb.so audit nodelay
session    required     pam_pwdb.so nodelay
password   required     pam_smbpass.so nodelay smbconf=/etc/samba.d/smb.conf
```

> **Note:** PAM allows stacking of authentication mechanisms. It is also possible to pass information obtained within one PAM module through to the next module in the PAM stack. Please refer to the documentation for your particular system implementation for details regarding the specific capabilities of PAM in this environment. Some Linux implmentations also provide the `pam_stack.so` module that allows all authentication to be configured in a single central file. The `pam_stack.so` method has some very devoted followers on the basis that it allows for easier administration. As with all issues in life though, every decision makes trade-offs, so you may want examine the PAM documentation for further helpful information.

19.2. Distributed Authentication

The astute administrator will realize from this that the combination of `pam_smbpass.so`, **winbindd**, and a distributed passdb backend, such as ldap, will allow the establishment of a centrally managed, distributed user/password database that can also be used by all PAM (eg: Linux) aware programs and applications. This arrangement can have particularly potent advantages compared with the use of Microsoft Active Directory Service (ADS) in so far as reduction of wide area network authentication traffic.

19.3. PAM Configuration in smb.conf

There is an option in smb.conf called obey pam restrictions (smb.conf.5.html#OBEYPAMRESTRICTIONS). The following is from the on-line help for this option in SWAT;

When Samba is configured to enable PAM support (i.e. `--with-pam`), this parameter will control whether or not Samba should obey PAM’s account and session management directives. The default behavior is to use PAM for clear text authentication only and to ignore any account or session management. Note that Samba always ignores PAM for authentication in the case of encrypt passwords = yes (smb.conf.5.html#ENCRYPTPASSWORDS). The reason is that PAM modules cannot support the challenge/response authentication mechanism needed in the presence of SMB password encryption.

Default: **obey pam restrictions = no**

Chapter 20. Stackable VFS modules

20.1. Introduction and configuration

Since samba 3.0, samba supports stackable VFS(Virtual File System) modules. Samba passes each request to access the unix file system thru the loaded VFS modules. This chapter covers all the modules that come with the samba source and references to some external modules.

You may have problems to compile these modules, as shared libraries are compiled and linked in different ways on different systems. They currently have been tested against GNU/linux and IRIX.

To use the VFS modules, create a share similar to the one below. The important parameter is the **vfs object** parameter which must point to the exact pathname of the shared library objects. For example, to log all access to files and use a recycle bin:

```
[audit]
        comment = Audited /data directory
        path = /data
        vfs object = /path/to/audit.so /path/to/recycle.so
        writeable = yes
        browseable = yes
```

The modules are used in the order they are specified.

Further documentation on writing VFS modules for Samba can be found in the Samba Developers Guide.

20.2. Included modules

20.2.1. audit

A simple module to audit file access to the syslog facility. The following operations are logged:

share
connect/disconnect
directory opens/create/remove
file open/close/rename/unlink/chmod

20.2.2. extd_audit

This module is identical with the *audit* module above except that it sends audit logs to both syslog as well as the smbd log file/s. The loglevel for this module is set in the smb.conf file. At loglevel = 0, only file and directory deletions and directory and file creations are logged. At loglevel = 1 file opens are renames and permission changes are logged , while at loglevel = 2 file open and close calls are logged also.

20.2.3. recycle

A recycle-bin like modules. When used any unlink call will be intercepted and files moved to the recycle directory instead of beeing deleted.

Supported options:

vfs_recycle_bin:repository

FIXME

vfs_recycle_bin:keeptree

FIXME

vfs_recycle_bin:versions

FIXME

vfs_recycle_bin:touch

FIXME

vfs_recycle_bin:maxsize

FIXME

vfs_recycle_bin:exclude

FIXME

vfs_recycle_bin:exclude_dir

FIXME

vfs_recycle_bin:noversions

FIXME

20.2.4. netatalk

A netatalk module, that will ease co-existence of samba and netatalk file sharing services.

Advantages compared to the old netatalk module:

it doesn't care about creating of .AppleDouble forks, just keeps ones in sync
if share in smb.conf doesn't contain .AppleDouble item in hide or veto list, it will be added automatically

20.3. VFS modules available elsewhere

This section contains a listing of various other VFS modules that have been posted but don't currently reside in the Samba CVS tree for one reason ot another (e.g. it is easy for the maintainer to have his or her own CVS tree).

No statemets about the stability or functionality any module should be implied due to its presence here.

20.3.1. DatabaseFS

URL: http://www.css.tayloru.edu/~elorimer/databasefs/index.php

By Eric Lorimer (mailto:elorimer@css.tayloru.edu).

I have created a VFS module which implements a fairly complete read-only filesystem. It presents information from a database as a filesystem in a modular and generic way to allow different databases to be used (originally designed for organizing MP3s under directories such as "Artists," "Song Keywords," etc... I have since applied it to a student roster database very easily). The directory structure is stored in the database itself and the module makes no assumptions about the database structure beyond the table it requires to run.

Any feedback would be appreciated: comments, suggestions, patches, etc... If nothing else, hopefully it might prove useful for someone else who wishes to create a virtual filesystem.

20.3.2. vscan

URL: http://www.openantivirus.org/

samba-vscan is a proof-of-concept module for Samba, which uses the VFS (virtual file system) features of Samba 2.2.x/3.0 alphaX. Of couse, Samba has to be compiled with VFS support. samba-vscan supports various virus scanners and is maintained by Rainer Link.

Chapter 21. Hosting a Microsoft Distributed File System tree on Samba

21.1. Instructions

The Distributed File System (or Dfs) provides a means of separating the logical view of files and directories that users see from the actual physical locations of these resources on the network. It allows for higher availability, smoother storage expansion, load balancing etc. For more information about Dfs, refer to Microsoft documentation (http://www.microsoft.com/NTServer/nts/downloads/winfeatures/NTSDistrFile/AdminGuide.asp).

This document explains how to host a Dfs tree on a Unix machine (for Dfs-aware clients to browse) using Samba.

To enable SMB-based DFS for Samba, configure it with the *`--with-msdfs`* option. Once built, a Samba server can be made a Dfs server by setting the global boolean *`host msdfs`* (smb.conf.5.html#HOSTMSDFS) parameter in the `smb.conf` file. You designate a share as a Dfs root using the share level boolean *`msdfs root`* (smb.conf.5.html#MSDFSROOT) parameter. A Dfs root directory on Samba hosts Dfs links in the form of symbolic links that point to other servers. For example, a symbolic link `junction->msdfs:storage1\share1` in the share directory acts as the Dfs junction. When Dfs-aware clients attempt to access the junction link, they are redirected to the storage location (in this case, \\storage1\share1).

Dfs trees on Samba work with all Dfs-aware clients ranging from Windows 95 to 2000.

Here's an example of setting up a Dfs tree on a Samba server.

```
# The smb.conf file:
[global]
netbios name = SAMBA
host msdfs   = yes

[dfs]
path = /export/dfsroot
msdfs root = yes
```

In the /export/dfsroot directory we set up our dfs links to other servers on the network.

```
root# cd /export/dfsroot
```

```
root# chown root /export/dfsroot
```

```
root# chmod 755 /export/dfsroot
```

```
root# ln -s msdfs:storageA\\shareA linka
```

```
root# ln -s msdfs:serverB\\share,serverC\\share linkb
```

You should set up the permissions and ownership of the directory acting as the Dfs root such that only designated users can create, delete or modify the msdfs links. Also note that symlink names should be all lowercase. This limitation exists to have Samba avoid trying all the case combinations to get at the link name. Finally set up the symbolic links to point to the network shares you want, and start Samba.

Users on Dfs-aware clients can now browse the Dfs tree on the Samba server at \\samba\dfs. Accessing links linka or linkb (which appear as directories to the client) takes users directly to the appropriate shares on the network.

21.1.1. Notes

- Windows clients need to be rebooted if a previously mounted non-dfs share is made a dfs root or vice versa. A better way is to introduce a new share and make it the dfs root.
- Currently there’s a restriction that msdfs symlink names should all be lowercase.
- For security purposes, the directory acting as the root of the Dfs tree should have ownership and permissions set so that only designated users can modify the symbolic links in the directory.

Chapter 22. Integrating MS Windows networks with Samba

This section deals with NetBIOS over TCP/IP name to IP address resolution. If you your MS Windows clients are NOT configured to use NetBIOS over TCP/IP then this section does not apply to your installation. If your installation involves use of NetBIOS over TCP/IP then this section may help you to resolve networking problems.

> **Note:** NetBIOS over TCP/IP has nothing to do with NetBEUI. NetBEUI is NetBIOS over Logical Link Control (LLC). On modern networks it is highly advised to NOT run NetBEUI at all. Note also that there is NO such thing as NetBEUI over TCP/IP - the existence of such a protocol is a complete and utter mis-apprehension.

Since the introduction of MS Windows 2000 it is possible to run MS Windows networking without the use of NetBIOS over TCP/IP. NetBIOS over TCP/IP uses UDP port 137 for NetBIOS name resolution and uses TCP port 139 for NetBIOS session services. When NetBIOS over TCP/IP is disabled on MS Windows 2000 and later clients then only TCP port 445 will be used and UDP port 137 and TCP port 139 will not.

> **Note:** When using Windows 2000 or later clients, if NetBIOS over TCP/IP is NOT disabled, then the client will use UDP port 137 (NetBIOS Name Service, also known as the Windows Internet Name Service or WINS), TCP port 139 AND TCP port 445 (for actual file and print traffic).

When NetBIOS over TCP/IP is disabled the use of DNS is essential. Most installations that disable NetBIOS over TCP/IP today use MS Active Directory Service (ADS). ADS requires Dynamic DNS with Service Resource Records (SRV RR) and with Incremental Zone Transfers (IXFR). Use of DHCP with ADS is recommended as a further means of maintaining central control over client workstation network configuration.

22.1. Name Resolution in a pure Unix/Linux world

The key configuration files covered in this section are:

- `/etc/hosts`
- `/etc/resolv.conf`
- `/etc/host.conf`
- `/etc/nsswitch.conf`

22.1.1. `/etc/hosts`

Contains a static list of IP Addresses and names. eg:

```
127.0.0.1 localhost localhost.localdomain
```

```
192.168.1.1 bigbox.caldera.com bigbox alias4box
```

The purpose of `/etc/hosts` is to provide a name resolution mechanism so that uses do not need to remember IP addresses.

Network packets that are sent over the physical network transport layer communicate not via IP addresses but rather using the Media Access Control address, or MAC address. IP Addresses are currently 32 bits in length and are typically presented as four (4) decimal numbers that are separated by a dot (or period). eg: 168.192.1.1

MAC Addresses use 48 bits (or 6 bytes) and are typically represented as two digit hexadecimal numbers separated by colons. eg: 40:8e:0a:12:34:56

Every network interfrace must have an MAC address. Associated with a MAC address there may be one or more IP addresses. There is NO relationship between an IP address and a MAC address, all such assignments are arbitary or discretionary in nature. At the most basic level all network communications takes place using MAC addressing. Since MAC addresses must be globally unique, and generally remains fixed for any particular interface, the assignment of an IP address makes sense from a network management perspective. More than one IP address can be assigned per MAC address. One address must be the primary IP address, this is the address that will be returned in the ARP reply.

When a user or a process wants to communicate with another machine the protocol implementation ensures that the "machine name" or "host name" is resolved to an IP address in a manner that is controlled by the TCP/IP configuration control files. The file `/etc/hosts` is one such file.

When the IP address of the destination interface has been determined a protocol called ARP/RARP is used to identify the MAC address of the target interface. ARP stands for Address Resolution Protocol, and is a broadcast oriented method that uses UDP (User Datagram Protocol) to send a request to all interfaces on the local network segment using the all 1's MAC address. Network interfaces are programmed to respond to two MAC addresses only; their own unique address and the address ff:ff:ff:ff:ff:ff. The reply packet from an ARP request will contain the MAC address and the primary IP address for each interface.

The `/etc/hosts` file is foundational to all Unix/Linux TCP/IP installations and as a minumum will contain the localhost and local network interface IP addresses and the primary names by which they are known within the local machine. This file helps to prime the pump so that a basic level of name resolution can exist before any other method of name resolution becomes available.

22.1.2. `/etc/resolv.conf`

This file tells the name resolution libraries:

- The name of the domain to which the machine belongs
- The name(s) of any domains that should be automatically searched when trying to resolve unqualified host names to their IP address

- The name or IP address of available Domain Name Servers that may be asked to perform name to address translation lookups

22.1.3. `/etc/host.conf`

`/etc/host.conf` is the primary means by which the setting in /etc/resolv.conf may be affected. It is a critical configuration file. This file controls the order by which name resolution may procede. The typical structure is:

```
order hosts,bind
multi on
```

then both addresses should be returned. Please refer to the man page for host.conf for further details.

22.1.4. `/etc/nsswitch.conf`

This file controls the actual name resolution targets. The file typically has resolver object specifications as follows:

```
# /etc/nsswitch.conf
#
# Name Service Switch configuration file.
#

passwd: compat
# Alternative entries for password authentication are:
# passwd: compat files nis ldap winbind
shadow: compat
group: compat

hosts: files nis dns
# Alternative entries for host name resolution are:
# hosts: files dns nis nis+ hesoid db compat ldap wins
networks: nis files dns

ethers: nis files
protocols: nis files
rpc: nis files
services: nis files
```

Of course, each of these mechanisms requires that the appropriate facilities and/or services are correctly configured.

It should be noted that unless a network request/message must be sent, TCP/IP networks are silent. All TCP/IP communications assumes a principal of speaking only when necessary.

Starting with version 2.2.0 samba has Linux support for extensions to the name service switch infrastructure so that linux clients will be able to obtain resolution of MS Windows NetBIOS names to IP Addresses. To gain this functionality Samba needs to be compiled with appropriate arguments to the make command (ie: **make nsswitch/libnss_wins.so**). The resulting library should then be installed in the `/lib` directory and the "wins" parameter needs to be added to the "hosts:" line in the `/etc/nsswitch.conf` file. At this point it will be possible to ping any MS Windows machine by it's NetBIOS machine name, so long as that machine is within the workgroup to which both the samba machine and the MS Windows machine belong.

22.2. Name resolution as used within MS Windows networking

MS Windows networking is predicated about the name each machine is given. This name is known variously (and inconsistently) as the "computer name", "machine name", "networking name", "netbios name", "SMB name". All terms mean the same thing with the exception of "netbios name" which can apply also to the name of the workgroup or the domain name. The terms "workgroup" and "domain" are really just a simply name with which the machine is associated. All NetBIOS names are exactly 16 characters in length. The 16th character is reserved. It is used to store a one byte value that indicates service level information for the NetBIOS name that is registered. A NetBIOS machine name is therefore registered for each service type that is provided by the client/server.

The following are typical NetBIOS name/service type registrations:

```
Unique NetBIOS Names:
MACHINENAME<00> = Server Service is running on MACHINENAME
MACHINENAME<03> = Generic Machine Name (NetBIOS name)
MACHINENAME<20> = LanMan Server service is running on MACHINENAME
WORKGROUP<1b> = Domain Master Browser

Group Names:
WORKGROUP<03> = Generic Name registered by all members of WORKGROUP
WORKGROUP<1c> = Domain Controllers / Netlogon Servers
WORKGROUP<1d> = Local Master Browsers
WORKGROUP<1e> = Internet Name Resolvers
```

It should be noted that all NetBIOS machines register their own names as per the above. This is in vast contrast to TCP/IP installations where traditionally the system administrator will determine in the /etc/hosts or in the DNS database what names are associated with each IP address.

One further point of clarification should be noted, the `/etc/hosts` file and the DNS records do not provide the NetBIOS name type information that MS Windows clients depend on to locate the type of service that may be needed. An example of this is what happens when an MS Windows client wants to locate a domain logon server.

It find this service and the IP address of a server that provides it by performing a lookup (via a NetBIOS broadcast) for enumeration of all machines that have registered the name type *<1c>. A logon request is then sent to each IP address that is returned in the enumerated list of IP addresses. Which ever machine first replies then ends up providing the logon services.

The name "workgroup" or "domain" really can be confusing since these have the added significance of indicating what is the security architecture of the MS Windows network. The term "workgroup" indicates that the primary nature of the network environment is that of a peer-to-peer design. In a WORKGROUP all machines are responsible for their own security, and generally such security is limited to use of just a password (known as SHARE MODE security). In most situations with peer-to-peer networking the users who control their own machines will simply opt to have no security at all. It is possible to have USER MODE security in a WORKGROUP environment, thus requiring use of a user name and a matching password.

MS Windows networking is thus predetermined to use machine names for all local and remote machine message passing. The protocol used is called Server Message Block (SMB) and this is implemented using the NetBIOS protocol (Network Basic Input Output System). NetBIOS can be encapsulated using LLC (Logical Link Control) protocol - in which case the resulting protocol is called NetBEUI (Network Basic Extended User Interface). NetBIOS can also be run over IPX (Internetworking Packet Exchange) protocol as used by Novell NetWare, and it can be run over TCP/IP protocols - in which case the resulting protocol is called NBT or NetBT, the NetBIOS over TCP/IP.

MS Windows machines use a complex array of name resolution mechanisms. Since we are primarily concerned with TCP/IP this demonstration is limited to this area.

22.2.1. The NetBIOS Name Cache

All MS Windows machines employ an in memory buffer in which is stored the NetBIOS names and IP addresses for all external machines that that machine has communicated with over the past 10-15 minutes. It is more efficient to obtain an IP address for a machine from the local cache than it is to go through all the configured name resolution mechanisms.

If a machine whose name is in the local name cache has been shut down before the name had been expired and flushed from the cache, then an attempt to exchange a message with that machine will be subject to time-out delays. i.e.: Its name is in the cache, so a name resolution lookup will succeed, but the machine can not respond. This can be frustrating for users - but it is a characteristic of the protocol.

The MS Windows utility that allows examination of the NetBIOS name cache is called "nbtstat". The Samba equivalent of this is called "nmblookup".

22.2.2. The LMHOSTS file

This file is usually located in MS Windows NT 4.0 or 2000 in `C:\WINNT\SYSTEM32\DRIVERS\ETC` and contains the IP Address and the machine name in matched pairs. The `LMHOSTS` file performs NetBIOS name to IP address mapping oriented.

It typically looks like:

```
# Copyright (c) 1998 Microsoft Corp.
#
# This is a sample LMHOSTS file used by the Microsoft Wins Client (NetBIOS
# over TCP/IP) stack for Windows98
#
# This file contains the mappings of IP addresses to NT computernames
# (NetBIOS) names.  Each entry should be kept on an individual line.
# The IP address should be placed in the first column followed by the
# corresponding computername. The address and the compter name
# should be separated by at least one space or tab. The "#" character
# is generally used to denote the start of a comment (see the exceptions
# below).
#
# This file is compatible with Microsoft LAN Manager 2.x TCP/IP lmhosts
# files and offers the following extensions:
#
#      #PRE
#      #DOM:<domain>
#      #INCLUDE <filename>
#      #BEGIN_ALTERNATE
#      #END_ALTERNATE
#      \0xnn (non-printing character support)
#
# Following any entry in the file with the characters "#PRE" will cause
# the entry to be preloaded into the name cache. By default, entries are
# not preloaded, but are parsed only after dynamic name resolution fails.
#
# Following an entry with the "#DOM:<domain>" tag will associate the
# entry with the domain specified by <domain>. This affects how the
# browser and logon services behave in TCP/IP environments. To preload
# the host name associated with #DOM entry, it is necessary to also add a
# #PRE to the line. The <domain> is always preloaded although it will not
# be shown when the name cache is viewed.
#
# Specifying "#INCLUDE <filename>" will force the RFC NetBIOS (NBT)
# software to seek the specified <filename> and parse it as if it were
# local. <filename> is generally a UNC-based name, allowing a
# centralized lmhosts file to be maintained on a server.
# It is ALWAYS necessary to provide a mapping for the IP address of the
# server prior to the #INCLUDE. This mapping must use the #PRE directive.
# In addtion the share "public" in the example below must be in the
# LanManServer list of "NullSessionShares" in order for client machines to
# be able to read the lmhosts file successfully. This key is under
# \machine\system\currentcontrolset\services\lanmanserver\parameters\nullsessionshares
# in the registry. Simply add "public" to the list found there.
#
# The #BEGIN_ and #END_ALTERNATE keywords allow multiple #INCLUDE
# statements to be grouped together. Any single successful include
# will cause the group to succeed.
#
# Finally, non-printing characters can be embedded in mappings by
# first surrounding the NetBIOS name in quotations, then using the
# \0xnn notation to specify a hex value for a non-printing character.
#
```

```
# The following example illustrates all of these extensions:
#
# 102.54.94.97     rhino         #PRE #DOM:networking  #net group's DC
# 102.54.94.102    "appname  \0x14"                    #special app server
# 102.54.94.123    popular            #PRE             #source server
# 102.54.94.117    localsrv           #PRE             #needed for the include
#
# #BEGIN_ALTERNATE
# #INCLUDE \\localsrv\public\lmhosts
# #INCLUDE \\rhino\public\lmhosts
# #END_ALTERNATE
#
# In the above example, the "appname" server contains a special
# character in its name, the "popular" and "localsrv" server names are
# preloaded, and the "rhino" server name is specified so it can be used
# to later #INCLUDE a centrally maintained lmhosts file if the "localsrv"
# system is unavailable.
#
# Note that the whole file is parsed including comments on each lookup,
# so keeping the number of comments to a minimum will improve performance.
# Therefore it is not advisable to simply add lmhosts file entries onto the
# end of this file.
```

22.2.3. HOSTS file

This file is usually located in MS Windows NT 4.0 or 2000 in `C:\WINNT\SYSTEM32\DRIVERS\ETC` and contains the IP Address and the IP hostname in matched pairs. It can be used by the name resolution infrastructure in MS Windows, depending on how the TCP/IP environment is configured. This file is in every way the equivalent of the Unix/Linux `/etc/hosts` file.

22.2.4. DNS Lookup

This capability is configured in the TCP/IP setup area in the network configuration facility. If enabled an elaborate name resolution sequence is followed the precise nature of which isdependant on what the NetBIOS Node Type parameter is configured to. A Node Type of 0 means use NetBIOS broadcast (over UDP broadcast) is first used if the name that is the subject of a name lookup is not found in the NetBIOS name cache. If that fails then DNS, HOSTS and LMHOSTS are checked. If set to Node Type 8, then a NetBIOS Unicast (over UDP Unicast) is sent to the WINS Server to obtain a lookup before DNS, HOSTS, LMHOSTS, or broadcast lookup is used.

22.2.5. WINS Lookup

A WINS (Windows Internet Name Server) service is the equivaent of the rfc1001/1002 specified NBNS (NetBIOS Name Server). A WINS server stores the names and IP addresses that are registered by a Windows client if the TCP/IP setup has been given at least one WINS Server IP Address.

To configure Samba to be a WINS server the following parameter needs to be added to the `smb.conf` file:

```
wins support = Yes
```

To configure Samba to use a WINS server the following parameters are needed in the `smb.conf` file:

```
wins support = No
wins server = xxx.xxx.xxx.xxx
```

where *`xxx.xxx.xxx.xxx`* is the IP address of the WINS server.

Chapter 23. Improved browsing in samba

23.1. Overview of browsing

SMB networking provides a mechanism by which clients can access a list of machines in a network, a so-called **browse list**. This list contains machines that are ready to offer file and/or print services to other machines within the network. Thus it does not include machines which aren't currently able to do server tasks. The browse list is heavily used by all SMB clients. Configuration of SMB browsing has been problematic for some Samba users, hence this document.

MS Windows 2000 and later, as with Samba 3 and later, can be configured to not use NetBIOS over TCP/IP. When configured this way it is imperative that name resolution (using DNS/LDAP/ADS) be correctly configured and operative. Browsing will NOT work if name resolution from SMB machine names to IP addresses does not function correctly.

Where NetBIOS over TCP/IP is enabled use of a WINS server is highly recommended to aid the resolution of NetBIOS (SMB) names to IP addresses. WINS allows remote segment clients to obtain NetBIOS name_type information that can NOT be provided by any other means of name resolution.

23.2. Browsing support in samba

Samba facilitates browsing. The browsing is supported by nmbd and is also controlled by options in the `smb.conf` file. Samba can act as a local browse master for a workgroup and the ability for samba to support domain logons and scripts is now available.

Samba can also act as a domain master browser for a workgroup. This means that it will collate lists from local browse masters into a wide area network server list. In order for browse clients to resolve the names they may find in this list, it is recommended that both samba and your clients use a WINS server.

Note that you should NOT set Samba to be the domain master for a workgroup that has the same name as an NT Domain: on each wide area network, you must only ever have one domain master browser per workgroup, regardless of whether it is NT, Samba or any other type of domain master that is providing this service.

> **Note:** Nmbd can be configured as a WINS server, but it is not necessary to specifically use samba as your WINS server. MS Windows NT4, Server or Advanced Server 2000 or 2003 can be configured as your WINS server. In a mixed NT/2000/2003 server and samba environment on a Wide Area Network, it is recommended that you use the Microsoft WINS server capabilities. In a samba-only environment, it is recommended that you use one and only one Samba server as your WINS server.

To get browsing to work you need to run nmbd as usual, but will need to use the **workgroup** option in `smb.conf` to control what workgroup Samba becomes a part of.

Samba also has a useful option for a Samba server to offer itself for browsing on another subnet. It is recommended that this option is only used for 'unusual' purposes: announcements over the internet, for example. See **remote announce** in the `smb.conf` man page.

23.3. Problem resolution

If something doesn't work then hopefully the log.nmb file will help you track down the problem. Try a debug level of 2 or 3 for finding problems. Also note that the current browse list usually gets stored in text form in a file called `browse.dat`.

Note that if it doesn't work for you, then you should still be able to type the server name as `\\SERVER` in filemanager then hit enter and filemanager should display the list of available shares.

Some people find browsing fails because they don't have the global **guest account** set to a valid account. Remember that the IPC$ connection that lists the shares is done as guest, and thus you must have a valid guest account.

MS Windows 2000 and upwards (as with Samba) can be configured to disallow anonymous (ie: Guest account) access to the IPC$ share. In that case, the MS Windows 2000/XP/2003 machine acting as an SMB/CIFS client will use the name of the currently logged in user to query the IPC$ share. MS Windows 9X clients are not able to do this and thus will NOT be able to browse server resources.

The other big problem people have is that their broadcast address, netmask or IP address is wrong (specified with the "interfaces" option in `smb.conf`)

23.4. Browsing across subnets

Since the release of Samba 1.9.17(alpha1) Samba has been updated to enable it to support the replication of browse lists across subnet boundaries. New code and options have been added to achieve this. This section describes how to set this feature up in different settings.

To see browse lists that span TCP/IP subnets (ie. networks separated by routers that don't pass broadcast traffic) you must set up at least one WINS server. The WINS server acts as a DNS for NetBIOS names, allowing NetBIOS name to IP address translation to be done by doing a direct query of the WINS server. This is done via a directed UDP packet on port 137 to the WINS server machine. The reason for a WINS server is that by default, all NetBIOS name to IP address translation is done by broadcasts from the querying machine. This means that machines on one subnet will not be able to resolve the names of machines on another subnet without using a WINS server.

Remember, for browsing across subnets to work correctly, all machines, be they Windows 95, Windows NT, or Samba servers must have the IP address of a WINS server given to them by a DHCP server, or by manual configuration (for Win95 and WinNT, this is in the TCP/IP Properties, under Network settings) for Samba this is in the `smb.conf` file.

23.4.1. How does cross subnet browsing work ?

Cross subnet browsing is a complicated dance, containing multiple moving parts. It has taken Microsoft several years to get the code that achieves this correct, and Samba lags behind in some areas. Samba is capable of cross subnet browsing when configured correctly.

Consider a network set up as follows :

```
                                  (DMB)
            N1_A      N1_B        N1_C       N1_D        N1_E
             |          |           |          |           |
         -------------------------------------------------------
           |          subnet 1                       |
         +---+                                      +---+
         |R1 | Router 1                  Router 2   |R2 |
         +---+                                      +---+
           |                                          |
           |  subnet 2              subnet 3          |
 --------------------------       ------------------------------------
 |     |     |      |               |        |         |           |
N2_A  N2_B  N2_C   N2_D           N3_A     N3_B      N3_C        N3_D
                   (WINS)
```

Consisting of 3 subnets (1, 2, 3) connected by two routers (R1, R2) - these do not pass broadcasts. Subnet 1 has 5 machines on it, subnet 2 has 4 machines, subnet 3 has 4 machines. Assume for the moment that all these machines are configured to be in the same workgroup (for simplicities sake). Machine N1_C on subnet 1 is configured as Domain Master Browser (ie. it will collate the browse lists for the workgroup). Machine N2_D is configured as WINS server and all the other machines are configured to register their NetBIOS names with it.

As all these machines are booted up, elections for master browsers will take place on each of the three subnets. Assume that machine N1_C wins on subnet 1, N2_B wins on subnet 2, and N3_D wins on subnet 3 - these machines are known as local master browsers for their particular subnet. N1_C has an advantage in winning as the local master browser on subnet 1 as it is set up as Domain Master Browser.

On each of the three networks, machines that are configured to offer sharing services will broadcast that they are offering these services. The local master browser on each subnet will receive these broadcasts and keep a record of the fact that the machine is offering a service. This list of records is the basis of the browse list. For this case, assume that all the machines are configured to offer services so all machines will be on the browse list.

For each network, the local master browser on that network is considered ’authoritative’ for all the names it receives via local broadcast. This is because a machine seen by the local master browser via a local broadcast must be on the same network as the local master browser and thus is a ’trusted’ and ’verifiable’ resource. Machines on other networks that the local master browsers learn about when collating their browse lists have not been directly seen - these records are called ’non-authoritative’.

At this point the browse lists look as follows (these are the machines you would see in your network neighborhood if you looked in it on a particular network right now).

```
Subnet           Browse Master   List
------           -------------   ----
Subnet1          N1_C            N1_A, N1_B, N1_C, N1_D, N1_E

Subnet2          N2_B            N2_A, N2_B, N2_C, N2_D

Subnet3          N3_D            N3_A, N3_B, N3_C, N3_D
```

Note that at this point all the subnets are separate, no machine is seen across any of the subnets.

Now examine subnet 2. As soon as N2_B has become the local master browser it looks for a Domain master browser to synchronize its browse list with. It does this by querying the WINS server (N2_D) for the IP address associated with the NetBIOS name WORKGROUP>1B<. This name was registerd by the Domain master browser (N1_C) with the WINS server as soon as it was booted.

Once N2_B knows the address of the Domain master browser it tells it that is the local master browser for subnet 2 by sending a MasterAnnouncement packet as a UDP port 138 packet. It then synchronizes with it by doing a NetServerEnum2 call. This tells the Domain Master Browser to send it all the server names it knows about. Once the domain master browser receives the MasterAnnouncement packet it schedules a synchronization request to the sender of that packet. After both synchronizations are done the browse lists look like :

```
Subnet           Browse Master   List
------           -------------   ----
Subnet1          N1_C            N1_A, N1_B, N1_C, N1_D, N1_E,
                                 N2_A(*), N2_B(*), N2_C(*), N2_D(*)

Subnet2          N2_B            N2_A, N2_B, N2_C, N2_D
                                 N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*)

Subnet3          N3_D            N3_A, N3_B, N3_C, N3_D

Servers with a (*) after them are non-authoritative names.
```

At this point users looking in their network neighborhood on subnets 1 or 2 will see all the servers on both, users on subnet 3 will still only see the servers on their own subnet.

The same sequence of events that occured for N2_B now occurs for the local master browser on subnet 3 (N3_D). When it synchronizes browse lists with the domain master browser (N1_A) it gets both the server entries on subnet 1, and those on subnet 2. After N3_D has synchronized with N1_C and vica-versa the browse lists look like.

```
Subnet           Browse Master   List
------           -------------   ----
Subnet1          N1_C            N1_A, N1_B, N1_C, N1_D, N1_E,
                                 N2_A(*), N2_B(*), N2_C(*), N2_D(*),
                                 N3_A(*), N3_B(*), N3_C(*), N3_D(*)

Subnet2          N2_B            N2_A, N2_B, N2_C, N2_D
                                 N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*)

Subnet3          N3_D            N3_A, N3_B, N3_C, N3_D
                                 N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*),
                                 N2_A(*), N2_B(*), N2_C(*), N2_D(*)

Servers with a (*) after them are non-authoritative names.
```

At this point users looking in their network neighborhood on subnets 1 or 3 will see all the servers on all sunbets, users on subnet 2 will still only see the servers on subnets 1 and 2, but not 3.

Finally, the local master browser for subnet 2 (N2_B) will sync again with the domain master browser (N1_C) and will recieve the missing server entries. Finally - and as a steady state (if no machines are removed or shut off) the browse lists will look like :

```
Subnet           Browse Master   List
------           -------------   ----
Subnet1          N1_C            N1_A, N1_B, N1_C, N1_D, N1_E,
                                 N2_A(*), N2_B(*), N2_C(*), N2_D(*),
                                 N3_A(*), N3_B(*), N3_C(*), N3_D(*)

Subnet2          N2_B            N2_A, N2_B, N2_C, N2_D
                                 N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*)
                                 N3_A(*), N3_B(*), N3_C(*), N3_D(*)

Subnet3          N3_D            N3_A, N3_B, N3_C, N3_D
                                 N1_A(*), N1_B(*), N1_C(*), N1_D(*), N1_E(*),
                                 N2_A(*), N2_B(*), N2_C(*), N2_D(*)

Servers with a (*) after them are non-authoritative names.
```

Synchronizations between the domain master browser and local master browsers will continue to occur, but this should be a steady state situation.

If either router R1 or R2 fails the following will occur:

1. Names of computers on each side of the inaccessible network fragments will be maintained for as long as 36 minutes, in the network neighbourhood lists.
2. Attempts to connect to these inaccessible computers will fail, but the names will not be removed from the network neighbourhood lists.

3. If one of the fragments is cut off from the WINS server, it will only be able to access servers on its local subnet, by using subnet-isolated broadcast NetBIOS name resolution. The effects are similar to that of losing access to a DNS server.

23.5. Setting up a WINS server

Either a Samba machine or a Windows NT Server machine may be set up as a WINS server. To set a Samba machine to be a WINS server you must add the following option to the `smb.conf` file on the selected machine : in the [globals] section add the line

wins support = yes

Versions of Samba prior to 1.9.17 had this parameter default to yes. If you have any older versions of Samba on your network it is strongly suggested you upgrade to a recent version, or at the very least set the parameter to 'no' on all these machines.

Machines with **wins support = yes** will keep a list of all NetBIOS names registered with them, acting as a DNS for NetBIOS names.

You should set up only ONE wins server. Do NOT set the **wins support = yes** option on more than one Samba server.

To set up a Windows NT Server as a WINS server you need to set up the WINS service - see your NT documentation for details. Note that Windows NT WINS Servers can replicate to each other, allowing more than one to be set up in a complex subnet environment. As Microsoft refuse to document these replication protocols Samba cannot currently participate in these replications. It is possible in the future that a Samba->Samba WINS replication protocol may be defined, in which case more than one Samba machine could be set up as a WINS server but currently only one Samba server should have the **wins support = yes** parameter set.

After the WINS server has been configured you must ensure that all machines participating on the network are configured with the address of this WINS server. If your WINS server is a Samba machine, fill in the Samba machine IP address in the "Primary WINS Server" field of the "Control Panel->Network->Protocols->TCP->WINS Server" dialogs in Windows 95 or Windows NT. To tell a Samba server the IP address of the WINS server add the following line to the [global] section of all `smb.conf` files :

wins server = >name or IP address<

where >name or IP address< is either the DNS name of the WINS server machine or its IP address.

Note that this line MUST NOT BE SET in the `smb.conf` file of the Samba server acting as the WINS server itself. If you set both the **wins support = yes** option and the **wins server = <name>** option then nmbd will fail to start.

There are two possible scenarios for setting up cross subnet browsing. The first details setting up cross subnet browsing on a network containing Windows 95, Samba and Windows NT machines that are not configured as part of a Windows NT Domain. The second details setting up cross subnet browsing on networks that contain NT Domains.

23.6. Setting up Browsing in a WORKGROUP

To set up cross subnet browsing on a network containing machines in up to be in a WORKGROUP, not an NT Domain you need to set up one Samba server to be the Domain Master Browser (note that this is *NOT* the same as a Primary Domain Controller, although in an NT Domain the same machine plays both roles). The role of a Domain master browser is to collate the browse lists from local master browsers on all the subnets that have a machine participating in the workgroup. Without one machine configured as a domain master browser each subnet would be an isolated workgroup, unable to see any machines on any other subnet. It is the presense of a domain master browser that makes cross subnet browsing possible for a workgroup.

In an WORKGROUP environment the domain master browser must be a Samba server, and there must only be one domain master browser per workgroup name. To set up a Samba server as a domain master browser, set the following option in the [global] section of the `smb.conf` file :

domain master = yes

The domain master browser should also preferrably be the local master browser for its own subnet. In order to achieve this set the following options in the [global] section of the `smb.conf` file :

```
domain master = yes
local master = yes
preferred master = yes
os level = 65
```

The domain master browser may be the same machine as the WINS server, if you require.

Next, you should ensure that each of the subnets contains a machine that can act as a local master browser for the workgroup. Any MS Windows NT/2K/XP/2003 machine should be able to do this, as will Windows 9x machines (although these tend to get rebooted more often, so it's not such a good idea to use these). To make a Samba server a local master browser set the following options in the [global] section of the `smb.conf` file :

```
domain master = no
local master = yes
preferred master = yes
os level = 65
```

Do not do this for more than one Samba server on each subnet, or they will war with each other over which is to be the local master browser.

The **local master** parameter allows Samba to act as a local master browser. The **preferred master** causes nmbd to force a browser election on startup and the **os level** parameter sets Samba high enough so that it should win any browser elections.

If you have an NT machine on the subnet that you wish to be the local master browser then you can disable Samba from becoming a local master browser by setting the following options in the **[global]** section of the `smb.conf` file :

```
domain master = no
local master = no
preferred master = no
os level = 0
```

23.7. Setting up Browsing in a DOMAIN

If you are adding Samba servers to a Windows NT Domain then you must not set up a Samba server as a domain master browser. By default, a Windows NT Primary Domain Controller for a Domain name is also the Domain master browser for that name, and many things will break if a Samba server registers the Domain master browser NetBIOS name (*`DOMAIN`*<1B>) with WINS instead of the PDC.

For subnets other than the one containing the Windows NT PDC you may set up Samba servers as local master browsers as described. To make a Samba server a local master browser set the following options in the **[global]** section of the `smb.conf` file :

```
domain master = no
local master = yes
preferred master = yes
os level = 65
```

If you wish to have a Samba server fight the election with machines on the same subnet you may set the **os level** parameter to lower levels. By doing this you can tune the order of machines that will become local master browsers if they are running. For more details on this see the section Forcing samba to be the master browser below.

If you have Windows NT machines that are members of the domain on all subnets, and you are sure they will always be running then you can disable Samba from taking part in browser elections and ever becoming a local master browser by setting following options in the **[global]** section of the `smb.conf` file :

domain master = no local master = no preferred master = no os level = 0

23.8. Forcing samba to be the master

Who becomes the **master browser** is determined by an election process using broadcasts. Each election packet contains a number of parameters which determine what precedence (bias) a host should have in the election. By default Samba uses a very low precedence and thus loses elections to just about anyone else.

If you want Samba to win elections then just set the **os level** global option in `smb.conf` to a higher number. It defaults to 0. Using 34 would make it win all elections over every other system (except other samba systems!)

A **os level** of 2 would make it beat WfWg and Win95, but not MS Windows NT/2K Server. A MS Windows NT/2K Server domain controller uses level 32.

The maximum os level is 255

If you want samba to force an election on startup, then set the **preferred master** global option in `smb.conf` to "yes". Samba will then have a slight advantage over other potential master browsers that are not preferred master browsers. Use this parameter with care, as if you have two hosts (whether they are windows 95 or NT or samba) on the same local subnet both set with **preferred master** to "yes", then periodically and continually they will force an election in order to become the local master browser.

If you want samba to be a **domain master browser**, then it is recommended that you also set **preferred master** to "yes", because samba will not become a domain master browser for the whole of your LAN or WAN if it is not also a local master browser on its own broadcast isolated subnet.

It is possible to configure two samba servers to attempt to become the domain master browser for a domain. The first server that comes up will be the domain master browser. All other samba servers will attempt to become the domain master browser every 5 minutes. They will find that another samba server is already the domain master browser and will fail. This provides automatic redundancy, should the current domain master browser fail.

23.9. Making samba the domain master

The domain master is responsible for collating the browse lists of multiple subnets so that browsing can occur between subnets. You can make samba act as the domain master by setting **domain master = yes** in `smb.conf`. By default it will not be a domain master.

Note that you should NOT set Samba to be the domain master for a workgroup that has the same name as an NT Domain.

When samba is the domain master and the master browser it will listen for master announcements (made roughly every twelve minutes) from local master browsers on other subnets and then contact them to synchronise browse lists.

If you want samba to be the domain master then I suggest you also set the **os level** high enough to make sure it wins elections, and set **preferred master** to "yes", to get samba to force an election on startup.

Note that all your servers (including samba) and clients should be using a WINS server to resolve NetBIOS names. If your clients are only using broadcasting to resolve NetBIOS names, then two things will occur:

1. your local master browsers will be unable to find a domain master browser, as it will only be looking on the local subnet.
2. if a client happens to get hold of a domain-wide browse list, and a user attempts to access a host in that list, it will be unable to resolve the NetBIOS name of that host.

If, however, both samba and your clients are using a WINS server, then:

1. your local master browsers will contact the WINS server and, as long as samba has registered that it is a domain master browser with the WINS server, your local master browser will receive samba's ip address as its domain master browser.
2. when a client receives a domain-wide browse list, and a user attempts to access a host in that list, it will contact the WINS server to resolve the NetBIOS name of that host. as long as that host has registered its NetBIOS name with the same WINS server, the user will be able to see that host.

23.10. Note about broadcast addresses

If your network uses a "0" based broadcast address (for example if it ends in a 0) then you will strike problems. Windows for Workgroups does not seem to support a 0's broadcast and you will probably find that browsing and name lookups won't work.

23.11. Multiple interfaces

Samba now supports machines with multiple network interfaces. If you have multiple interfaces then you will need to use the **interfaces** option in `smb.conf` to configure them.

Chapter 24. Securing Samba

24.1. Introduction

This note was attached to the Samba 2.2.8 release notes as it contained an important security fix. The information contained here applies to Samba installations in general.

24.2. Using host based protection

In many installations of Samba the greatest threat comes for outside your immediate network. By default Samba will accept connections from any host, which means that if you run an insecure version of Samba on a host that is directly connected to the Internet you can be especially vulnerable.

One of the simplest fixes in this case is to use the **hosts allow** and **hosts deny** options in the Samba `smb.conf` configuration file to only allow access to your server from a specific range of hosts. An example might be:

```
hosts allow = 127.0.0.1 192.168.2.0/24 192.168.3.0/24
hosts deny = 0.0.0.0/0
```

The above will only allow SMB connections from 'localhost' (your own computer) and from the two private networks 192.168.2 and 192.168.3. All other connections will be refused connections as soon as the client sends its first packet. The refusal will be marked as a 'not listening on called name' error.

24.3. Using interface protection

By default Samba will accept connections on any network interface that it finds on your system. That means if you have a ISDN line or a PPP connection to the Internet then Samba will accept connections on those links. This may not be what you want.

You can change this behaviour using options like the following:

```
interfaces = eth* lo
bind interfaces only = yes
```

This tells Samba to only listen for connections on interfaces with a name starting with 'eth' such as eth0, eth1, plus on the loopback interface called 'lo'. The name you will need to use depends on what OS you are using, in the above I used the common name for Ethernet adapters on Linux.

If you use the above and someone tries to make a SMB connection to your host over a PPP interface called 'ppp0' then they will get a TCP connection refused reply. In that case no Samba code is run at all as the operating system has been told not to pass connections from that interface to any process.

24.4. Using a firewall

Many people use a firewall to deny access to services that they don't want exposed outside their network. This can be a very good idea, although I would recommend using it in conjunction with the above methods so that you are protected even if your firewall is not active for some reason.

If you are setting up a firewall then you need to know what TCP and UDP ports to allow and block. Samba uses the following:

```
UDP/137    - used by nmbd
UDP/138    - used by nmbd
TCP/139    - used by smbd
TCP/445    - used by smbd
```

The last one is important as many older firewall setups may not be aware of it, given that this port was only added to the protocol in recent years.

24.5. Using a IPC$ share deny

If the above methods are not suitable, then you could also place a more specific deny on the IPC$ share that is used in the recently discovered security hole. This allows you to offer access to other shares while denying access to IPC$ from potentially untrustworthy hosts.

To do that you could use:

```
  [ipc$]
     hosts allow = 192.168.115.0/24 127.0.0.1
     hosts deny = 0.0.0.0/0
```

this would tell Samba that IPC$ connections are not allowed from anywhere but the two listed places (localhost and a local subnet). Connections to other shares would still be allowed. As the IPC$ share is the only share that is always accessible anonymously this provides some level of protection against attackers that do not know a username/password for your host.

If you use this method then clients will be given a ’access denied’ reply when they try to access the IPC$ share. That means that those clients will not be able to browse shares, and may also be unable to access some other resources.

This is not recommended unless you cannot use one of the other methods listed above for some reason.

24.6. Upgrading Samba

Please check regularly on http://www.samba.org/ for updates and important announcements. Occasionally security releases are made and it is highly recommended to upgrade Samba when a security vulnerability is discovered.

Chapter 25. Unicode/Charsets

25.1. What are charsets and unicode?

Computers communicate in numbers. In texts, each number will be translated to a corresponding letter. The meaning that will be assigned to a certain number depends on the *character set(charset)* that is used. A charset can be seen as a table that is used to translate numbers to letters. Not all computers use the same charset (there are charsets with German umlauts, Japanese characters, etc). Usually a charset contains 256 characters, which means that storing a character with it takes exactly one byte.

There are also charsets that support even more characters, but those need twice(or even more) as much storage space. These charsets can contain **256 * 256 = 65536** characters, which is more then all possible characters one could think of. They are called multibyte charsets (because they use more then one byte to store one character).

A standardised multibyte charset is unicode, info available at www.unicode.org (http://www.unicode.org/). Big advantage of using a multibyte charset is that you only need one; no need to make sure two computers use the same charset when they are communicating.

Old windows clients used to use single-byte charsets, named 'codepages' by microsoft. However, there is no support for negotiating the charset to be used in the smb protocol. Thus, you have to make sure you are using the same charset when talking to an old client. Newer clients (Windows NT, 2K, XP) talk unicode over the wire.

25.2. Samba and charsets

As of samba 3.0, samba can (and will) talk unicode over the wire. Internally, samba knows of three kinds of character sets:

unix charset

> This is the charset used internally by your operating system. The default is *ASCII*, which is fine for most systems.

display charset

> This is the charset samba will use to print messages on your screen. It should generally be the same as the **unix charset**.

dos charset

> This is the charset samba uses when communicating with DOS and Windows 9x clients. It will talk unicode to all newer clients. The default depends on the charsets you have installed on your system. Run **testparm -v | grep "dos charset"** to see what the default is on your system.

IV. Appendixes

Chapter 26. SWAT - The Samba Web Admininistration Tool

This is a rough guide to SWAT.

26.1. SWAT Features and Benefits

You must use at least the following ...

26.1.1. The SWAT Home Page

Blah blah here.

26.1.2. Global Settings

Document steps right here!

26.1.3. The SWAT Wizard

Lots of blah blah here.

26.1.4. Share Settings

Document steps right here!

26.1.5. Printing Settings

Document steps right here!

26.1.6. The Status Page

Document steps right here!

26.1.7. The Password Change Page

Document steps right here!

Chapter 27. Migration from NT4 PDC to Samba-3 PDC

This is a rough guide to assist those wishing to migrate from NT4 domain control to Samba-3 based domain control.

27.1. Planning and Getting Started

You must use at least the following ...

27.1.1. Objectives

Blah blah objectives here.

27.1.2. Steps In Migration Process

Document steps right here!

27.2. Managing Samba-3 Domain Control

Lots of blah blah here.

Chapter 28. Samba performance issues

28.1. Comparisons

The Samba server uses TCP to talk to the client. Thus if you are trying to see if it performs well you should really compare it to programs that use the same protocol. The most readily available programs for file transfer that use TCP are ftp or another TCP based SMB server.

If you want to test against something like a NT or WfWg server then you will have to disable all but TCP on either the client or server. Otherwise you may well be using a totally different protocol (such as Netbeui) and comparisons may not be valid.

Generally you should find that Samba performs similarly to ftp at raw transfer speed. It should perform quite a bit faster than NFS, although this very much depends on your system.

Several people have done comparisons between Samba and Novell, NFS or WinNT. In some cases Samba performed the best, in others the worst. I suspect the biggest factor is not Samba vs some other system but the hardware and drivers used on the various systems. Given similar hardware Samba should certainly be competitive in speed with other systems.

28.2. Socket options

There are a number of socket options that can greatly affect the performance of a TCP based server like Samba.

The socket options that Samba uses are settable both on the command line with the -O option, or in the smb.conf file.

The "socket options" section of the smb.conf manual page describes how to set these and gives recommendations.

Getting the socket options right can make a big difference to your performance, but getting them wrong can degrade it by just as much. The correct settings are very dependent on your local network.

The socket option TCP_NODELAY is the one that seems to make the biggest single difference for most networks. Many people report that adding "socket options = TCP_NODELAY" doubles the read performance of a Samba drive. The best explanation I have seen for this is that the Microsoft TCP/IP stack is slow in sending tcp ACKs.

28.3. Read size

The option "read size" affects the overlap of disk reads/writes with network reads/writes. If the amount of data being transferred in several of the SMB commands (currently SMBwrite, SMBwriteX and SMBreadbraw) is larger than this value then the server begins writing the data before it has received the whole packet from the network, or in the case of SMBreadbraw, it begins writing to the network before all the data has been read from disk.

This overlapping works best when the speeds of disk and network access are similar, having very little effect when the speed of one is much greater than the other.

The default value is 16384, but very little experimentation has been done yet to determine the optimal value, and it is likely that the best value will vary greatly between systems anyway. A value over 65536 is pointless and will cause you to allocate memory unnecessarily.

28.4. Max xmit

At startup the client and server negotiate a "maximum transmit" size, which limits the size of nearly all SMB commands. You can set the maximum size that Samba will negotiate using the "max xmit = " option in smb.conf. Note that this is the maximum size of SMB request that Samba will accept, but not the maximum size that the *client* will accept. The client maximum receive size is sent to Samba by the client and Samba honours this limit.

It defaults to 65536 bytes (the maximum), but it is possible that some clients may perform better with a smaller transmit unit. Trying values of less than 2048 is likely to cause severe problems.

In most cases the default is the best option.

28.5. Log level

If you set the log level (also known as "debug level") higher than 2 then you may suffer a large drop in performance. This is because the server flushes the log file after each operation, which can be very expensive.

28.6. Read raw

The "read raw" operation is designed to be an optimised, low-latency file read operation. A server may choose to not support it, however. and Samba makes support for "read raw" optional, with it being enabled by default.

In some cases clients don’t handle "read raw" very well and actually get lower performance using it than they get using the conventional read operations.

So you might like to try "read raw = no" and see what happens on your network. It might lower, raise or not affect your performance. Only testing can really tell.

28.7. Write raw

The "write raw" operation is designed to be an optimised, low-latency file write operation. A server may choose to not support it, however. and Samba makes support for "write raw" optional, with it being enabled by default.

Some machines may find "write raw" slower than normal write, in which case you may wish to change this option.

28.8. Slow Clients

One person has reported that setting the protocol to COREPLUS rather than LANMAN2 gave a dramatic speed improvement (from 10k/s to 150k/s).

I suspect that his PC's (386sx16 based) were asking for more data than they could chew. I suspect a similar speed could be had by setting "read raw = no" and "max xmit = 2048", instead of changing the protocol. Lowering the "read size" might also help.

28.9. Slow Logins

Slow logins are almost always due to the password checking time. Using the lowest practical "password level" will improve things a lot. You could also enable the "UFC crypt" option in the Makefile.

28.10. Client tuning

Often a speed problem can be traced to the client. The client (for example Windows for Workgroups) can often be tuned for better TCP performance.

See your client docs for details. In particular, I have heard rumours that the WfWg options TCPWINDOWSIZE and TCPSEGMENTSIZE can have a large impact on performance.

Also note that some people have found that setting DefaultRcvWindow in the [MSTCP] section of the SYSTEM.INI file under WfWg to 3072 gives a big improvement. I don't know why.

My own experience wth DefaultRcvWindow is that I get much better performance with a large value (16384 or larger). Other people have reported that anything over 3072 slows things down enourmously. One person even reported a speed drop of a factor of 30 when he went from 3072 to 8192. I don't know why.

It probably depends a lot on your hardware, and the type of unix box you have at the other end of the link.

Paul Cochrane has done some testing on client side tuning and come to the following conclusions:

Install the W2setup.exe file from www.microsoft.com. This is an update for the winsock stack and utilities which improve performance.

Configure the win95 TCPIP registry settings to give better perfomance. I use a program called MTUSPEED.exe which I got off the net. There are various other utilities of this type freely available. The setting which give the best performance for me are:

1. MaxMTU Remove
2. RWIN Remove
3. MTUAutoDiscover Disable
4. MTUBlackHoleDetect Disable
5. Time To Live Enabled
6. Time To Live - HOPS 32
7. NDI Cache Size 0

I tried virtually all of the items mentioned in the document and the only one which made a difference to me was the socket options. It turned out I was better off without any!!!!!

In terms of overall speed of transfer, between various win95 clients and a DX2-66 20MB server with a crappy NE2000 compatible and old IDE drive (Kernel 2.0.30). The transfer rate was reasonable for 10 baseT.

```
The figures are:          Put              Get
P166 client 3Com card:    420-440kB/s      500-520kB/s
P100 client 3Com card:    390-410kB/s      490-510kB/s
DX4-75 client NE2000:     370-380kB/s      330-350kB/s
```

I based these test on transfer two files a 4.5MB text file and a 15MB textfile. The results arn't bad considering the hardware Samba is running on. It's a crap machine!!!!

The updates mentioned in 1 and 2 brought up the transfer rates from just over 100kB/s in some clients.

A new client is a P333 connected via a 100MB/s card and hub. The transfer rates from this were good: 450-500kB/s on put and 600+kB/s on get.

Looking at standard FTP throughput, Samba is a bit slower (100kB/s upwards). I suppose there is more going on in the samba protocol, but if it could get up to the rate of FTP the perfomance would be quite staggering.

Chapter 29. Portability

Samba works on a wide range of platforms but the interface all the platforms provide is not always compatible. This chapter contains platform-specific information about compiling and using samba.

29.1. HPUX

HP's implementation of supplementary groups is, er, non-standard (for hysterical reasons). There are two group files, /etc/group and /etc/logingroup; the system maps UIDs to numbers using the former, but initgroups() reads the latter. Most system admins who know the ropes symlink /etc/group to /etc/logingroup (hard link doesn't work for reasons too stupid to go into here). initgroups() will complain if one of the groups you're in in /etc/logingroup has what it considers to be an invalid ID, which means outside the range [0..UID_MAX], where UID_MAX is (I think) 60000 currently on HP-UX. This precludes -2 and 65534, the usual 'nobody' GIDs.

If you encounter this problem, make sure that the programs that are failing to initgroups() be run as users not in any groups with GIDs outside the allowed range.

This is documented in the HP manual pages under setgroups(2) and passwd(4).

On HPUX you must use gcc or the HP Ansi compiler. The free compiler that comes with HP-UX is not Ansi compliant and cannot compile Samba.

29.2. SCO Unix

If you run an old version of SCO Unix then you may need to get important TCP/IP patches for Samba to work correctly. Without the patch, you may encounter corrupt data transfers using samba.

The patch you need is UOD385 Connection Drivers SLS. It is available from SCO (ftp.sco.com, directory SLS, files uod385a.Z and uod385a.ltr.Z).

29.3. DNIX

DNIX has a problem with seteuid() and setegid(). These routines are needed for Samba to work correctly, but they were left out of the DNIX C library for some reason.

For this reason Samba by default defines the macro NO_EID in the DNIX section of includes.h. This works around the problem in a limited way, but it is far from ideal, some things still won't work right.

To fix the problem properly you need to assemble the following two functions and then either add them to your C library or link them into Samba.

put this in the file `setegid.s`:

```
        .globl  _setegid
_setegid:
        moveq   #47,d0
        movl    #100,a0
        moveq   #1,d1
        movl    4(sp),a1
        trap    #9
        bccs    1$
        jmp     cerror
1$:
        clrl    d0
        rts
```

put this in the file `seteuid.s`:

```
        .globl  _seteuid
_seteuid:
        moveq   #47,d0
        movl    #100,a0
        moveq   #0,d1
        movl    4(sp),a1
        trap    #9
        bccs    1$
        jmp     cerror
1$:
        clrl    d0
        rts
```

after creating the above files you then assemble them using

as seteuid.s

as setegid.s

that should produce the files `seteuid.o` and `setegid.o`

then you need to add these to the LIBSM line in the DNIX section of the Samba Makefile. Your LIBSM line will then look something like this:

```
LIBSM = setegid.o seteuid.o -ln
```

You should then remove the line:

```
#define NO_EID
```

from the DNIX section of `includes.h`

29.4. RedHat Linux Rembrandt-II

By default RedHat Rembrandt-II during installation adds an entry to /etc/hosts as follows:

```
127.0.0.1 loopback "hostname"."domainname"
```

This causes Samba to loop back onto the loopback interface. The result is that Samba fails to communicate correctly with the world and therefor may fail to correctly negotiate who is the master browse list holder and who is the master browser.

Corrective Action: Delete the entry after the word loopback in the line starting 127.0.0.1

29.5. AIX

29.5.1. Sequential Read Ahead

Disabling Sequential Read Ahead using **`vmtune -r 0`** improves samba performance significally.

Chapter 30. Samba and other CIFS clients

This chapter contains client-specific information.

30.1. Macintosh clients?

Yes. Thursby (http://www.thursby.com/) now have a CIFS Client / Server called DAVE - see

They test it against Windows 95, Windows NT and samba for compatibility issues. At the time of writing, DAVE was at version 1.0.1. The 1.0.0 to 1.0.1 update is available as a free download from the Thursby web site (the speed of finder copies has been greatly enhanced, and there are bug-fixes included).

Alternatives - There are two free implementations of AppleTalk for several kinds of UNIX machnes, and several more commercial ones. These products allow you to run file services and print services natively to Macintosh users, with no additional support required on the Macintosh. The two free omplementations are Netatalk (http://www.umich.edu/~rsug/netatalk/), and CAP (http://www.cs.mu.oz.au/appletalk/atalk.html). What Samba offers MS Windows users, these packages offer to Macs. For more info on these packages, Samba, and Linux (and other UNIX-based systems) see http://www.eats.com/linux_mac_win.html

30.2. OS2 Client

30.2.1. How can I configure OS/2 Warp Connect or OS/2 Warp 4 as a client for Samba?

A more complete answer to this question can be found on http://carol.wins.uva.nl/~leeuw/samba/warp.html (http://carol.wins.uva.nl/~leeuw/samba/warp.html).

Basically, you need three components:

- The File and Print Client ('IBM Peer')
- TCP/IP ('Internet support')
- The "NetBIOS over TCP/IP" driver ('TCPBEUI')

Installing the first two together with the base operating system on a blank system is explained in the Warp manual. If Warp has already been installed, but you now want to install the networking support, use the "Selective Install for Networking" object in the "System Setup" folder.

Adding the "NetBIOS over TCP/IP" driver is not described in the manual and just barely in the online documentation. Start MPTS.EXE, click on OK, click on "Configure LAPS" and click on "IBM OS/2 NETBIOS OVER TCP/IP" in 'Protocols'. This line is then moved to 'Current Configuration'. Select that line, click on "Change number" and increase it from 0 to 1. Save this configuration.

If the Samba server(s) is not on your local subnet, you can optionally add IP names and addresses of these servers to the "Names List", or specify a WINS server ('NetBIOS Nameserver' in IBM and RFC terminology). For Warp Connect you may need to download an update for 'IBM Peer' to bring it on the same level as Warp 4. See the webpage mentioned above.

30.2.2. How can I configure OS/2 Warp 3 (not Connect), OS/2 1.2, 1.3 or 2.x for Samba?

You can use the free Microsoft LAN Manager 2.2c Client for OS/2 from ftp://ftp.microsoft.com/BusSys/Clients/LANMAN.OS2/ (ftp://ftp.microsoft.com/BusSys/Clients/LANMAN.OS2/). See http://carol.wins.uva.nl/~leeuw/lanman.html (http://carol.wins.uva.nl/~leeuw/lanman.html) for more information on how to install and use this client. In a nutshell, edit the file \OS2VER in the root directory of the OS/2 boot partition and add the lines:

```
20=setup.exe
20=netwksta.sys
20=netvdd.sys
```

before you install the client. Also, don't use the included NE2000 driver because it is buggy. Try the NE2000 or NS2000 driver from ftp://ftp.cdrom.com/pub/os2/network/ndis/ (ftp://ftp.cdrom.com/pub/os2/network/ndis/) instead.

30.2.3. Are there any other issues when OS/2 (any version) is used as a client?

When you do a NET VIEW or use the "File and Print Client Resource Browser", no Samba servers show up. This can be fixed by a patch from http://carol.wins.uva.nl/~leeuw/samba/fix.html (http://carol.wins.uva.nl/~leeuw/samba/fix.html). The patch will be included in a later version of Samba. It also fixes a couple of other problems, such as preserving long filenames when objects are dragged from the Workplace Shell to the Samba server.

30.2.4. How do I get printer driver download working for OS/2 clients?

First, create a share called [PRINTDRV] that is world-readable. Copy your OS/2 driver files there. Note that the .EA_ files must still be separate, so you will need to use the original install files, and not copy an installed driver from an OS/2 system.

Install the NT driver first for that printer. Then, add to your smb.conf a parameter, os2 driver map = `filename`". Then, in the file specified by `filename`, map the name of the NT driver name to the OS/2 driver name as follows:

nt driver name = os2 "driver name"."device name", e.g.: HP LaserJet 5L = LASERJET.HP LaserJet 5L

You can have multiple drivers mapped in this file.

If you only specify the OS/2 driver name, and not the device name, the first attempt to download the driver will actually download the files, but the OS/2 client will tell you the driver is not available. On the second attempt, it will work. This is fixed simply by adding the device name to the mapping, after which it will work on the first attempt.

30.3. Windows for Workgroups

30.3.1. Use latest TCP/IP stack from Microsoft

Use the latest TCP/IP stack from microsoft if you use Windows for workgroups.

The early TCP/IP stacks had lots of bugs.

Microsoft has released an incremental upgrade to their TCP/IP 32-Bit VxD drivers. The latest release can be found on their ftp site at ftp.microsoft.com, located in /peropsys/windows/public/tcpip/wfwt32.exe. There is an update.txt file there that describes the problems that were fixed. New files include WINSOCK.DLL, TELNET.EXE, WSOCK.386, VNBT.386, WSTCP.386, TRACERT.EXE, NETSTAT.EXE, and NBTSTAT.EXE.

30.3.2. Delete .pwl files after password change

WfWg does a lousy job with passwords. I find that if I change my password on either the unix box or the PC the safest thing to do is to delete the .pwl files in the windows directory. The PC will complain about not finding the files, but will soon get over it, allowing you to enter the new password.

If you don't do this you may find that WfWg remembers and uses the old password, even if you told it a new one.

Often WfWg will totally ignore a password you give it in a dialog box.

30.3.3. Configure WfW password handling

There is a program call admincfg.exe on the last disk (disk 8) of the WFW 3.11 disk set. To install it type EXPAND A:\ADMINCFG.EX_ C:\WINDOWS\ADMINCFG.EXE Then add an icon for it via the "Progam Manager" "New" Menu. This program allows you to control how WFW handles passwords. ie disable Password Caching etc for use with **security = user**

30.3.4. Case handling of passwords

Windows for Workgroups uppercases the password before sending it to the server. Unix passwords can be case-sensitive though. Check the smb.conf(5) (smb.conf.5.html) information on **password level** to specify what characters samba should try to uppercase when checking.

30.3.5. Use TCP/IP as default protocol

To support print queue reporting you may find that you have to use TCP/IP as the default protocol under WfWg. For some reason if you leave Netbeui as the default it may break the print queue reporting on some systems. It is presumably a WfWg bug.

30.4. Windows ’95/’98

When using Windows 95 OEM SR2 the following updates are recommended where Samba is being used. Please NOTE that the above change will affect you once these updates have been installed.

There are more updates than the ones mentioned here. You are referred to the Microsoft Web site for all currently available updates to your specific version of Windows 95.

1. Kernel Update: KRNLUPD.EXE
2. Ping Fix: PINGUPD.EXE
3. RPC Update: RPCRTUPD.EXE
4. TCP/IP Update: VIPUPD.EXE
5. Redirector Update: VRDRUPD.EXE

Also, if using MS OutLook it is desirable to install the OLEUPD.EXE fix. This fix may stop your machine from hanging for an extended period when exiting OutLook and you may also notice a significant speedup when accessing network neighborhood services.

30.5. Windows 2000 Service Pack 2

There are several annoyances with Windows 2000 SP2. One of which only appears when using a Samba server to host user profiles to Windows 2000 SP2 clients in a Windows domain. This assumes that Samba is a member of the domain, but the problem will likely occur if it is not.

In order to server profiles successfully to Windows 2000 SP2 clients (when not operating as a PDC), Samba must have **nt acl support = no** added to the file share which houses the roaming profiles. If this is not done, then the Windows 2000 SP2 client will complain about not being able to access the profile (Access Denied) and create multiple copies of it on disk (DOMAIN.user.001, DOMAIN.user.002, etc...). See the smb.conf(5) (smb.conf.5.html) man page for more details on this option. Also note that the **nt acl support** parameter was formally a global parameter in releases prior to Samba 2.2.2.

The following is a minimal profile share:

```
[profile]
path = /export/profile
create mask = 0600
directory mask = 0700
nt acl support = no
read only = no
```

The reason for this bug is that the Win2k SP2 client copies the security descriptor for the profile which contains the Samba server's SID, and not the domain SID. The client compares the SID for SAMBA\user and realizes it is different that the one assigned to DOMAIN\user. Hence the reason for the "access denied" message.

By disabling the **nt acl support** parameter, Samba will send the Win2k client a response to the QuerySecurityDescriptor trans2 call which causes the client to set a default ACL for the profile. This default ACL includes

DOMAIN\user "Full Control"

> **Note:** This bug does not occur when using winbind to create accounts on the Samba host for Domain users.

30.6. Windows NT 3.1

If you have problems communicating across routers with Windows NT 3.1 workstations, read this Microsoft Knowledge Base article (http://support.microsoft.com/default.aspx?scid=kb;[LN];Q103765).

Chapter 31. How to compile SAMBA

You can obtain the samba source from the samba website (http://samba.org/). To obtain a development version, you can download samba from CVS or using rsync.

31.1. Access Samba source code via CVS

31.1.1. Introduction

Samba is developed in an open environment. Developers use CVS (Concurrent Versioning System) to "checkin" (also known as "commit") new source code. Samba's various CVS branches can be accessed via anonymous CVS using the instructions detailed in this chapter.

This chapter is a modified version of the instructions found at http://samba.org/samba/cvs.html

31.1.2. CVS Access to samba.org

The machine samba.org runs a publicly accessible CVS repository for access to the source code of several packages, including samba, rsync and jitterbug. There are two main ways of accessing the CVS server on this host.

31.1.2.1. Access via CVSweb

You can access the source code via your favourite WWW browser. This allows you to access the contents of individual files in the repository and also to look at the revision history and commit logs of individual files. You can also ask for a diff listing between any two versions on the repository.

Use the URL : http://samba.org/cgi-bin/cvsweb

31.1.2.2. Access via cvs

You can also access the source code via a normal cvs client. This gives you much more control over you can do with the repository and allows you to checkout whole source trees and keep them up to date via normal cvs commands. This is the preferred method of access if you are a developer and not just a casual browser.

To download the latest cvs source code, point your browser at the URL : http://www.cyclic.com/. and click on the 'How to get cvs' link. CVS is free software under the GNU GPL (as is Samba). Note that there are several graphical CVS clients which provide a graphical interface to the sometimes mundane CVS commands. Links to theses clients are also available from http://www.cyclic.com.

To gain access via anonymous cvs use the following steps. For this example it is assumed that you want a copy of the samba source code. For the other source code repositories on this system just substitute the correct package name

1. Install a recent copy of cvs. All you really need is a copy of the cvs client binary.
2. Run the command

   ```
   cvs -d :pserver:cvs@samba.org:/cvsroot login
   ```

   When it asks you for a password type **`cvs`**.

3. Run the command

   ```
   cvs -d :pserver:cvs@samba.org:/cvsroot co samba
   ```

   This will create a directory called samba containing the latest samba source code (i.e. the HEAD tagged cvs branch). This currently corresponds to the 3.0 development tree.

   CVS branches other HEAD can be obtained by using the *`-r`* and defining a tag name. A list of branch tag names can be found on the "Development" page of the samba web site. A common request is to obtain the latest 2.2 release code. This could be done by using the following userinput.

   ```
   cvs -d :pserver:cvs@samba.org:/cvsroot co -r SAMBA_2_2 samba
   ```

4. Whenever you want to merge in the latest code changes use the following command from within the samba directory:

   ```
   cvs update -d -P
   ```

31.2. Accessing the samba sources via rsync and ftp

pserver.samba.org also exports unpacked copies of most parts of the CVS tree at ftp://pserver.samba.org/pub/unpacked and also via anonymous rsync at rsync://pserver.samba.org/ftp/unpacked/. I recommend using rsync rather than ftp. See the rsync homepage (http://rsync.samba.org/) for more info on rsync.

The disadvantage of the unpacked trees is that they do not support automatic merging of local changes like CVS does. rsync access is most convenient for an initial install.

31.3. Building the Binaries

To do this, first run the program **`./configure`** in the source directory. This should automatically configure Samba for your operating system. If you have unusual needs then you may wish to run

```
root# ./configure --help
```

first to see what special options you can enable. Then executing

```
root# make
```

will create the binaries. Once it's successfully compiled you can use

```
root# make install
```

to install the binaries and manual pages. You can separately install the binaries and/or man pages using

```
root# make installbin
```

and

```
root# make installman
```

Note that if you are upgrading for a previous version of Samba you might like to know that the old versions of the binaries will be renamed with a ".old" extension. You can go back to the previous version with

```
root# make revert
```

if you find this version a disaster!

31.3.1. Compiling samba with Active Directory support

In order to compile samba with ADS support, you need to have installed on your system:

the MIT kerberos development libraries (either install from the sources or use a package). The heimdal libraries will not work.
the OpenLDAP development libraries.

If your kerberos libraries are in a non-standard location then remember to add the configure option
--with-krb5=DIR.

After you run configure make sure that `include/config.h` it generates contains lines like this:

```
#define HAVE_KRB5 1
#define HAVE_LDAP 1
```

If it doesn't then configure did not find your krb5 libraries or your ldap libraries. Look in config.log to figure out why and fix it.

31.3.1.1. Installing the required packages for Debian

On Debian you need to install the following packages:

libkrb5-dev
krb5-user

31.3.1.2. Installing the required packages for RedHat

On RedHat this means you should have at least:

krb5-workstation (for kinit)
krb5-libs (for linking with)
krb5-devel (because you are compiling from source)

in addition to the standard development environment.

Note that these are not standard on a RedHat install, and you may need to get them off CD2.

31.4. Starting the smbd and nmbd

You must choose to start smbd and nmbd either as daemons or from inetdDon't try to do both! Either you can put them in `inetd.conf` and have them started on demand by inetd, or you can start them as daemons either from the command line or in `/etc/rc.local`. See the man pages for details on the command line options. Take particular care to read the bit about what user you need to be in order to start Samba. In many cases you must be root.

The main advantage of starting smbd and nmbd using the recommended daemon method is that they will respond slightly more quickly to an initial connection request.

31.4.1. Starting from inetd.conf

NOTE; The following will be different if you use NIS, NIS+ or LDAP to distribute services maps.

Look at your `/etc/services`. What is defined at port 139/tcp. If nothing is defined then add a line like this:

```
netbios-ssn 139/tcp
```

similarly for 137/udp you should have an entry like:

```
netbios-ns 137/udp
```

Next edit your `/etc/inetd.conf` and add two lines something like this:

```
netbios-ssn stream tcp nowait root /usr/local/samba/bin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/bin/nmbd nmbd
```

The exact syntax of `/etc/inetd.conf` varies between unixes. Look at the other entries in inetd.conf for a guide.

> **Note:** Some unixes already have entries like netbios_ns (note the underscore) in `/etc/services`. You must either edit `/etc/services` or `/etc/inetd.conf` to make them consistent.

> **Note:** On many systems you may need to use the **interfaces** option in `smb.conf` to specify the IP address and netmask of your interfaces. Run ifconfig as root if you don't know what the broadcast is for your net. nmbd tries to determine it at run time, but fails on some unixes.

> **Warning**
>
> Many unixes only accept around 5 parameters on the command line in `inetd.conf`. This means you shouldn't use spaces between the options and arguments, or you should use a script, and start the script from **inetd**.

Restart **inetd**, perhaps just send it a HUP. If you have installed an earlier version of nmbd then you may need to kill nmbd as well.

31.4.2. Alternative: starting it as a daemon

To start the server as a daemon you should create a script something like this one, perhaps calling it `startsmb`.

```
#!/bin/sh
/usr/local/samba/bin/smbd -D
/usr/local/samba/bin/nmbd -D
```

then make it executable with **chmod +x startsmb**

You can then run **startsmb** by hand or execute it from `/etc/rc.local`

To kill it send a kill signal to the processes **nmbd** and **smbd**.

> **Note:** If you use the SVR4 style init system then you may like to look at the `examples/svr4-startup` script to make Samba fit into that system.

Chapter 32. Reporting Bugs

32.1. Introduction

The email address for bug reports for stable releases is samba@samba.org (mailto:samba@samba.org). Bug reports for alpha releases should go to samba-technical@samba.org (mailto:samba-technical@samba.org).

Please take the time to read this file before you submit a bug report. Also, please see if it has changed between releases, as we may be changing the bug reporting mechanism at some time.

Please also do as much as you can yourself to help track down the bug. Samba is maintained by a dedicated group of people who volunteer their time, skills and efforts. We receive far more mail about it than we can possibly answer, so you have a much higher chance of an answer and a fix if you send us a "developer friendly" bug report that lets us fix it fast.

Do not assume that if you post the bug to the comp.protocols.smb newsgroup or the mailing list that we will read it. If you suspect that your problem is not a bug but a configuration problem then it is better to send it to the Samba mailing list, as there are (at last count) 5000 other users on that list that may be able to help you.

You may also like to look though the recent mailing list archives, which are conveniently accessible on the Samba web pages at http://samba.org/samba/.

32.2. General info

Before submitting a bug report check your config for silly errors. Look in your log files for obvious messages that tell you that you've misconfigured something and run testparm to test your config file for correct syntax.

Have you run through the diagnosis? This is very important.

If you include part of a log file with your bug report then be sure to annotate it with exactly what you were doing on the client at the time, and exactly what the results were.

32.3. Debug levels

If the bug has anything to do with Samba behaving incorrectly as a server (like refusing to open a file) then the log files will probably be very useful. Depending on the problem a log level of between 3 and 10 showing the problem may be appropriate. A higher level givesmore detail, but may use too much disk space.

To set the debug level use **log level =** in your `smb.conf`. You may also find it useful to set the log level higher for just one machine and keep separate logs for each machine. To do this use:

```
log level = 10
log file = /usr/local/samba/lib/log.%m
include = /usr/local/samba/lib/smb.conf.%m
```

then create a file `/usr/local/samba/lib/smb.conf.`*`machine`* where *`machine`* is the name of the client you wish to debug. In that file put any `smb.conf` commands you want, for example **log level=** may be useful. This also allows you to experiment with different security systems, protocol levels etc on just one machine.

The `smb.conf` entry **log level =** is synonymous with the entry **debuglevel =** that has been used in older versions of Samba and is being retained for backwards compatibility of `smb.conf` files.

As the **log level =** value is increased you will record a significantly increasing level of debugging information. For most debugging operations you may not need a setting higher than 3. Nearly all bugs can be tracked at a setting of 10, but be prepared for a VERY large volume of log data.

32.4. Internal errors

If you get a "INTERNAL ERROR" message in your log files it means that Samba got an unexpected signal while running. It is probably a segmentation fault and almost certainly means a bug in Samba (unless you have faulty hardware or system software).

If the message came from smbd then it will probably be accompanied by a message which details the last SMB message received by smbd. This info is often very useful in tracking down the problem so please include it in your bug report.

You should also detail how to reproduce the problem, if possible. Please make this reasonably detailed.

You may also find that a core file appeared in a `corefiles` subdirectory of the directory where you keep your samba log files. This file is the most useful tool for tracking down the bug. To use it you do this:

gdb smbd core

adding appropriate paths to smbd and core so gdb can find them. If you don't have gdb then try **`dbx`**. Then within the debugger use the command **`where`** to give a stack trace of where the problem occurred. Include this in your mail.

If you known any assembly language then do a **`disass`** of the routine where the problem occurred (if its in a library routine then disassemble the routine that called it) and try to work out exactly where the problem is by looking at the surrounding code. Even if you don't know assembly then incuding this info in the bug report can be useful.

32.5. Attaching to a running process

Unfortunately some unixes (in particular some recent linux kernels) refuse to dump a core file if the task has changed uid (which smbd does often). To debug with this sort of system you could try to attach to the running process using **`gdb smbd PID`** where you get *`PID`* from smbstatus. Then use **`c`** to continue and try to cause the core dump using the client. The debugger should catch the fault and tell you where it occurred.

32.6. Patches

The best sort of bug report is one that includes a fix! If you send us patches please use **`diff -u`** format if your version of diff supports it, otherwise use **`diff -c4`**. Make sure your do the diff against a clean version of the source and let me know exactly what version you used.

Chapter 33. The samba checklist

33.1. Introduction

This file contains a list of tests you can perform to validate your Samba server. It also tells you what the likely cause of the problem is if it fails any one of these steps. If it passes all these tests then it is probably working fine.

You should do ALL the tests, in the order shown. We have tried to carefully choose them so later tests only use capabilities verified in the earlier tests.

If you send one of the samba mailing lists an email saying "it doesn't work" and you have not followed this test procedure then you should not be surprised your email is ignored.

33.2. Assumptions

In all of the tests it is assumed you have a Samba server called BIGSERVER and a PC called ACLIENT both in workgroup TESTGROUP.

The procedure is similar for other types of clients.

It is also assumed you know the name of an available share in your `smb.conf`. I will assume this share is called *`tmp`*. You can add a *`tmp`* share like by adding the following to `smb.conf`:

```
[tmp]
 comment = temporary files
 path = /tmp
 read only = yes
```

> **Note:** These tests assume version 3.0 or later of the samba suite. Some commands shown did not exist in earlier versions.

Please pay attention to the error messages you receive. If any error message reports that your server is being unfriendly you should first check that you IP name resolution is correctly set up. eg: Make sure your `/etc/resolv.conf` file points to name servers that really do exist.

Also, if you do not have DNS server access for name resolution please check that the settings for your `smb.conf` file results in **dns proxy = no**. The best way to check this is with **`testparm smb.conf`**.

33.3. The tests

Diagnosing your samba server

1. In the directory in which you store your `smb.conf` file, run the command **`testparm smb.conf`**. If it reports any errors then your `smb.conf` configuration file is faulty.

   Note: Your `smb.conf` file may be located in: `/etc/samba` Or in: `/usr/local/samba/lib`

2. Run the command **`ping BIGSERVER`** from the PC and **`ping ACLIENT`** from the unix box. If you don't get a valid response then your TCP/IP software is not correctly installed.

   Note that you will need to start a "dos prompt" window on the PC to run ping.

   If you get a message saying "host not found" or similar then your DNS software or `/etc/hosts` file is not correctly setup. It is possible to run samba without DNS entries for the server and client, but I assume you do have correct entries for the remainder of these tests.

   Another reason why ping might fail is if your host is running firewall software. You will need to relax the rules to let in the workstation in question, perhaps by allowing access from another subnet (on Linux this is done via the ipfwadm program.)
3. Run the command **`smbclient -L BIGSERVER`** on the unix box. You should get a list of available shares back.

   If you get a error message containing the string "Bad password" then you probably have either an incorrect **hosts allow**, **hosts deny** or **valid users** line in your `smb.conf`, or your guest account is not valid. Check what your guest account is using testparm and temporarily remove any **hosts allow**, **hosts deny**, **valid users** or **invalid users** lines.

   If you get a "connection refused" response then the smbd server may not be running. If you installed it in inetd.conf then you probably edited that file incorrectly. If you installed it as a daemon then check that it is running, and check that the netbios-ssn port is in a LISTEN state using **`netstat -a`**.

   If you get a "session request failed" then the server refused the connection. If it says "Your server software is being unfriendly" then its probably because you have invalid command line parameters to smbd, or a similar fatal problem with the initial startup of smbd. Also check your config file (`smb.conf`) for syntax errors with testparm and that the various directories where samba keeps its log and lock files exist.

   There are a number of reasons for which smbd may refuse or decline a session request. The most common of these involve one or more of the following `smb.conf` file entries:

   ```
   hosts deny = ALL
   hosts allow = xxx.xxx.xxx.xxx/yy
   bind interfaces only = Yes
   ```

In the above, no allowance has been made for any session requests that will automatically translate to the loopback adaptor address 127.0.0.1. To solve this problem change these lines to:

```
hosts deny = ALL
hosts allow = xxx.xxx.xxx.xxx/yy 127.
```

Do NOT use the **bind interfaces only** parameter where you may wish to use the samba password change facility, or where smbclient may need to access local service for name resolution or for local resource connections. (Note: the **bind interfaces only** parameter deficiency where it will not allow connections to the loopback address will be fixed soon).

Another common cause of these two errors is having something already running on port 139, such as Samba (ie: smbd is running from inetd already) or something like Digital's Pathworks. Check your `inetd.conf` file before trying to start smbd as a daemon, it can avoid a lot of frustration!

And yet another possible cause for failure of this test is when the subnet mask and / or broadcast address settings are incorrect. Please check that the network interface IP Address / Broadcast Address / Subnet Mask settings are correct and that Samba has correctly noted these in the `log.nmb` file.

4. Run the command **`nmblookup -B BIGSERVER __SAMBA__`**. You should get the IP address of your Samba server back.

   If you don't then nmbd is incorrectly installed. Check your `inetd.conf` if you run it from there, or that the daemon is running and listening to udp port 137.

   One common problem is that many inetd implementations can't take many parameters on the command line. If this is the case then create a one-line script that contains the right parameters and run that from inetd.

5. run the command **`nmblookup -B ACLIENT '*'`**

   You should get the PCs IP address back. If you don't then the client software on the PC isn't installed correctly, or isn't started, or you got the name of the PC wrong.

   If ACLIENT doesn't resolve via DNS then use the IP address of the client in the above test.

6. Run the command **`nmblookup -d 2 '*'`**

   This time we are trying the same as the previous test but are trying it via a broadcast to the default broadcast address. A number of Netbios/TCPIP hosts on the network should respond, although Samba may not catch all of the responses in the short time it listens. You should see "got a positive name query response" messages from several hosts.

   If this doesn't give a similar result to the previous test then nmblookup isn't correctly getting your broadcast address through its automatic mechanism. In this case you should experiment use the **interfaces** option in `smb.conf` to manually configure your IP address, broadcast and netmask.

If your PC and server aren't on the same subnet then you will need to use the `-B` option to set the broadcast address to the that of the PCs subnet.

This test will probably fail if your subnet mask and broadcast address are not correct. (Refer to TEST 3 notes above).

7. Run the command **`smbclient //BIGSERVER/TMP`**. You should then be prompted for a password. You should use the password of the account you are logged into the unix box with. If you want to test with another account then add the *`-U accountname`* option to the end of the command line. eg: **`smbclient //bigserver/tmp -Ujohndoe`**

   > **Note:** It is possible to specify the password along with the username as follows: **`smbclient //bigserver/tmp -Ujohndoe%secret`**

   Once you enter the password you should get the `smb>` prompt. If you don't then look at the error message. If it says "invalid network name" then the service "tmp" is not correctly setup in your `smb.conf`.

   If it says "bad password" then the likely causes are:

   1. you have shadow passords (or some other password system) but didn't compile in support for them in smbd
   2. your **valid users** configuration is incorrect
   3. you have a mixed case password and you haven't enabled the **password level** option at a high enough level
   4. the **path =** line in `smb.conf` is incorrect. Check it with testparm
   5. you enabled password encryption but didn't create the SMB encrypted password file

   Once connected you should be able to use the commands **dir get put** etc. Type **help *`command`*** for instructions. You should especially check that the amount of free disk space shown is correct when you type **dir**.

8. On the PC type the command **`net view \\BIGSERVER`**. You will need to do this from within a "dos prompt" window. You should get back a list of available shares on the server.

   If you get a "network name not found" or similar error then netbios name resolution is not working. This is usually caused by a problem in nmbd. To overcome it you could do one of the following (you only need to choose one of them):

   1. fixup the nmbd installation
   2. add the IP address of BIGSERVER to the **wins server** box in the advanced tcp/ip setup on the PC.
   3. enable windows name resolution via DNS in the advanced section of the tcp/ip setup
   4. add BIGSERVER to your lmhosts file on the PC.

   If you get a "invalid network name" or "bad password error" then the same fixes apply as they did for the **`smbclient -L`** test above. In particular, make sure your **hosts allow** line is correct (see the man pages)

Also, do not overlook that fact that when the workstation requests the connection to the samba server it will attempt to connect using the name with which you logged onto your Windows machine. You need to make sure that an account exists on your Samba server with that exact same name and password.

If you get "specified computer is not receiving requests" or similar it probably means that the host is not contactable via tcp services. Check to see if the host is running tcp wrappers, and if so add an entry in the `hosts.allow` file for your client (or subnet, etc.)

9. Run the command **`net use x: \\BIGSERVER\TMP`**. You should be prompted for a password then you should get a "command completed successfully" message. If not then your PC software is incorrectly installed or your smb.conf is incorrect. make sure your **hosts allow** and other config lines in `smb.conf` are correct.

   It's also possible that the server can't work out what user name to connect you as. To see if this is the problem add the line **user = *`username`*** to the **[tmp]** section of `smb.conf` where *`username`* is the username corresponding to the password you typed. If you find this fixes things you may need the username mapping option.

   It might also be the case that your client only sends encrypted passwords and you have **encrypt passwords = no** in `smb.conf` Turn it back on to fix.

10. Run the command **`nmblookup -M`** ***`testgroup`*** where *`testgroup`* is the name of the workgroup that your Samba server and Windows PCs belong to. You should get back the IP address of the master browser for that workgroup.

    If you don't then the election process has failed. Wait a minute to see if it is just being slow then try again. If it still fails after that then look at the browsing options you have set in `smb.conf`. Make sure you have **preferred master = yes** to ensure that an election is held at startup.

11. From file manager try to browse the server. Your samba server should appear in the browse list of your local workgroup (or the one you specified in smb.conf). You should be able to double click on the name of the server and get a list of shares. If you get a "invalid password" error when you do then you are probably running WinNT and it is refusing to browse a server that has no encrypted password capability and is in user level security mode. In this case either set **security = server** AND **password server = Windows_NT_Machine** in your `smb.conf` file, or make sure **encrypted passwords** is set to "yes".

33.4. Still having troubles?

Try the mailing list or newsgroup, or use the ethereal utility to sniff the problem. The official samba mailing list can be reached at samba@samba.org (mailto:samba@samba.org). To find out more about samba and how to subscribe to the mailing list check out the samba web page at http://samba.org/samba/ (http://samba.org/samba)