Chapter 8

UPDATING SAMBA-3

It was a little difficult to select an appropriate title for this chapter. From email messages on the Samba mailing lists it is clear that many people consider the updating and upgrading of Samba to be a migration matter. Others talk about migrating Samba servers when in fact the issue at hand is one of installing a new Samba server to replace an older existing Samba server.

There has also been much talk about migration of Samba-3 from an smbpasswd passdb backend to the use of the tdbsam or ldapsam facilities that are new to Samba-3.

Clearly, there is not a great deal of clarity in the terminology that various people apply to these modes by which Samba servers are updated. This is further highlighted by an email posting that included the following neat remark:

> I like the "net rpc vampire" on NT4, but that to my surprise does not seem to work against a Samba PDC and, if addressed in the Samba to Samba context in either book, I could not find it.

So in response to the significant request for these situations to be better documented, this chapter has now been added. User contributions and documentation of real-world experiences are a most welcome addition to this chapter.

8.1 Introduction

A Windows network administrator explained in an email what changes he was planning to make and followed with the question: "Anyone done this before?" Many of us have upgraded and updated Samba without incident. Others have experienced much pain and user frustration. So it is to be hoped that the notes in this chapter will make a positive difference by assuring that someone will be saved a lot of discomfort.

Before anyone commences an upgrade or an update of Samba, the one cardinal rule that must be observed is: Backup all Samba configuration files in case it is necessary to revert to the old version. Even if you do not like this precautionary step, users will punish an administrator who fails to take adequate steps to avoid situations that may inflict lost productivity on them.

WARNING

Samba makes it possible to upgrade and update configuration files, but it is not possible to downgrade the configuration files. Please ensure that all configuration and control files are backed up to permit a downgrade in the rare event that this may be necessary.

It is prudent also to backup all data files on the server before attempting to perform a major upgrade. Many administrators have experienced the consequences of failure to take adequate precautions. So what is adequate? That is simple! If data is lost during an upgrade or update and it can not be restored, the precautions taken were inadequate. If a backup was not needed, but was available, caution was on the side of the victor.

8.1.1 Cautions and Notes

Someone once said, "It is good to be sorry, but better never to need to be!" These are wise words of advice to those contemplating a Samba upgrade or update.

This is as good a time as any to define the terms `upgrade` and `update`. The term `upgrade` refers to the installation of a version of Samba that is a whole generation or more ahead of that which is installed. Generations are indicated by the first digit of the version number. So far Samba has been released in generations 1.x, 2.x, 3.x, and currently 4.0 is in development.

The term `update` refers to a minor version number installation in place of one of the same generation. For example, updating from Samba 3.0.10 to 3.0.14 is an update. The move from Samba 2.0.7 to 3.0.14 is an upgrade.

While the use of these terms is an exercise in semantics, what needs to be realized is that there are major functional differences between a Samba 2.x release and a Samba 3.0.x release. Such differences may require a significantly different approach to solving the same networking challenge and generally require careful review of the latest documentation to identify precisely how the new installation may need to be modified to preserve prior functionality.

There is an old axiom that says, "The greater the volume of the documentation, the greater the risk that noone will read it, but where there is no documentation, noone can read it!" While true, some documentation is an evil necessity. It is hoped that this update to the documentation will avoid both extremes.

8.1.1.1 Security Identifiers (SIDs)

Before the days of Windows NT and OS/2, every Windows and DOS networking client that used the SMB protocols was an entirely autonomous entity. There was no concept of a security identifier for a machine or a user outside of the username, the machine name, and

the workgroup name. In actual fact, these were not security identifiers in the same context as the way that the SID is used since the development of Windows NT 3.10.

Versions of Samba prior to 1.9 did not make use of a SID. Instead they make exclusive use of the username that is embedded in the SessionSetUpAndX component of the connection setup process between a Windows client and an SMB/CIFS server.

Around November 1997 support was added to Samba-1.9 to handle the Windows security RPC-based protocols that implemented support for Samba to store a machine SID. This information was stored in a file called `MACHINE.SID`.

Within the lifetime of the early Samba 2.x series, the machine SID information was relocated into a tdb file called `secrets.tdb`, which is where it is still located in Samba 3.0.x along with other information that pertains to the local machine and its role within a domain security context.

There are two types of SID, those pertaining to the machine itself and the domain to which it may belong, and those pertaining to users and groups within the security context of the local machine, in the case of standalone servers (SAS) and domain member servers (DMS).

When the Samba **smbd** daemon is first started, if the `secrets.tdb` file does not exist, it is created at the first client connection attempt. If this file does exist, **smbd** checks that there is a machine SID (if it is a domain controller, it searches for the domain SID). If **smbd** does not find one for the current name of the machine or for the current name of the workgroup, a new SID will be generated and then written to the `secrets.tdb` file. The SID is generated in a nondeterminative manner. This means that each time it is generated for a particular combination of machine name (hostname) and domain name (workgroup), it will be different.

The SID is the key used by MS Windows networking for all networking operations. This means that when the machine or domain SID changes, all security-encoded objects such as profiles and ACLs may become unusable.

NOTE

It is of paramount importance that the machine and domain SID be backed up so that in the event of a change of hostname (machine name) or domain name (workgroup) the SID can be restored to its previous value.

In Samba-3 on a domain controller (PDC or BDC), the domain name controls the domain SID. On all prior versions the hostname (computer name, or NetBIOS name) controlled the SID. On a standalone server the hostname still controls the SID.

The local machine SID can be backed up using this procedure (Samba-3):

```
root#  net getlocalsid > /etc/samba/my-local-SID
```

The contents of the file `/etc/samba/my-local-SID` will be:

```
SID for domain FRODO is: S-1-5-21-726309263-4128913605-1168186429
```

This SID can be restored by executing:

```
root#  net setlocalsid S-1-5-21-726309263-4128913605-1168186429
```

Samba 1.9.x stored the machine SID in the the file `/etc/MACHINE.SID` from which it could be recovered and stored into the `secrets.tdb` file using the procedure shown above.

Where the `secrets.tdb` file exists and a version of Samba 2.x or later has been used, there is no specific need to go through this update process. Samba-3 has the ability to read the older tdb file and to perform an in-situ update to the latest tdb format. This is not a reversible process — it is a one-way upgrade.

In the course of the Samba 2.0.x series the **smbpasswd** was modified to permit the domain SID to be captured to the `secrets.tdb` file by executing:

```
root#  smbpasswd -S PDC -Uadministrator%password
```

The release of the Samba 2.2.x series permitted the SID to be obtained by executing:

```
root#  smbpasswd -S PDC -Uadministrator%password
```

from which the SID could be copied to a file and then written to the Samba-2.2.x `secrets.tdb` file by executing:

```
root#  smbpasswd -W S-1-5-21-726309263-4128913605-1168186429
```

Domain security information, which includes the domain SID, can be obtained from Samba-2.2.x systems by executing:

```
root#  rpcclient hostname lsaquery -Uroot%password
```

This can also be done with Samba-3 by executing:

```
root#  net rpc info -Uroot%password
```

```
Domain Name: MIDEARTH
Domain SID: S-1-5-21-726309263-4128913605-1168186429
Sequence number: 1113415916
Num users: 4237
Num domain groups: 86
Num local groups: 0
```

It is a very good practice to store this SID information in a safely kept file, just in case it is ever needed at a later date.

Take note that the domain SID is used extensively in Samba. Where LDAP is used for the *passdb backend*, all user, group, and trust accounts are encoded with the domain SID. This means that if the domain SID changes for any reason, the entire Samba environment can become broken and require extensive corrective action if the original SID cannot be restored. Fortunately, it can be recovered from a dump of the LDAP database. A dump of the LDAP directory database can be obtained by executing:

```
root#  slapcat -v -l filename.ldif
```

When the domain SID has changed, roaming profiles cease to be functional. The recovery of roaming profiles necessitates resetting of the domain portion of the user SID that owns the profile. This is encoded in the `NTUser.DAT` and can be updated using the Samba **profiles** utility. Please be aware that not all Linux distributions of the Samba RPMs include this essential utility. Please do not complain to the Samba Team if this utility is missing; that issue that must be addressed to the creator of the RPM package. The Samba Team do their best to make available all the tools needed to manage a Samba-based Windows networking environment.

8.1.1.2 Change of hostname

Samba uses two methods by which the primary NetBIOS machine name (also known as a computer name or the hostname) may be determined: If the `smb.conf` file contains a *netbios name* entry, its value will be used directly. In the absence of such an entry, the UNIX system hostname will be used.

Many sites have become victims of lost Samba functionality because the UNIX system hostname was changed for one reason or another. Such a change will cause a new machine SID to be generated. If this happens on a domain controller, it will also change the domain SID. These SIDs can be updated (restored) using the procedure outlined previously.

NOTE

Do NOT change the hostname or the *netbios name*. If this is changed, be sure to reset the machine SID to the original setting. Otherwise there may be serious interoperability and/or operational problems.

8.1.1.3 Change of Workgroup (Domain) Name

The domain name of a Samba server is identical to the workgroup name and is set in the `smb.conf` file using the *workgroup* parameter. This has been consistent throughout the history of Samba and across all versions.

Be aware that when the workgroup name is changed, a new SID will be generated. The old domain SID can be reset using the procedure outlined earlier in this chapter.

8.1.1.4 Location of config files

The Samba-Team has maintained a constant default location for all Samba control files throughout the life of the project. People who have produced binary packages of Samba have varied the location of the Samba control files. This has led to some confusion for network administrators.

The Samba 1.9.x `smb.conf` file may be found either in the `/etc` directory or in `/usr/local/samba/lib`.

During the life of the Samba 2.x release, the `smb.conf` file was relocated on Linux systems to the `/etc/samba` directory where it remains located also for Samba 3.0.x installations.

Samba 2.x introduced the `secrets.tdb` file that is also stored in the `/etc/samba` directory, or in the `/usr/local/samba/lib` directory subsystem.

The location at which **smbd** expects to find all configuration and control files is determined at the time of compilation of Samba. For versions of Samba prior to 3.0, one way to find the expected location of these files is to execute:

```
root#  strings /usr/sbin/smbd | grep conf
root#  strings /usr/sbin/smbd | grep secret
root#  strings /usr/sbin/smbd | grep smbpasswd
```

Note: The **smbd** executable may be located in the path `/usr/local/samba/sbin`.

Samba-3 provides a neat new way to track the location of all control files as well as to find the compile-time options used as the Samba package was built. Here is how the dark secrets of the internals of the location of control files within Samba executables can be uncovered:

```
root#  smbd -b | less
Build environment:
   Built by:    root@frodo
   Built on:    Mon Apr 11 20:23:27 MDT 2005
   Built using: gcc
   Build host:  Linux frodo 2.6...
   SRCDIR:      /usr/src/packages/BUILD/samba-3.0.20/source
   BUILDDIR:    /usr/src/packages/BUILD/samba-3.0.20/source

Paths:
   SBINDIR: /usr/sbin
   BINDIR: /usr/bin
   SWATDIR: /usr/share/samba/swat
   CONFIGFILE: /etc/samba/smb.conf
   LOGFILEBASE: /var/log/samba
   LMHOSTSFILE: /etc/samba/lmhosts
   LIBDIR: /usr/lib/samba
   SHLIBEXT: so
   LOCKDIR: /var/lib/samba
   PIDDIR: /var/run/samba
   SMB_PASSWD_FILE: /etc/samba/smbpasswd
   PRIVATE_DIR: /etc/samba
 ...
```

It is important that both the `smb.conf` file and the `secrets.tdb` be backed up before attempting any upgrade. The `secrets.tdb` file is version-encoded, and therefore a newer version may not work with an older version of Samba. A backup means that it is always possible to revert a failed or problematic upgrade.

8.1.1.5 International Language Support

Samba-2.x had no support for Unicode; instead, all national language character-set support in file names was done using particular locale codepage mapping techniques. Samba-3 supports Unicode in file names, thus providing true internationalization support.

Non-English users whose national language character set has special characters and who upgrade naively will find that many files that have the special characters in the file name will see them garbled and jumbled up. This typically happens with umlauts and accents because these characters were particular to the codepage that was in use with Samba-2.x using an 8-bit encoding scheme.

Files that are created with Samba-3 will use UTF-8 encoding. Should the file system ever end up with a mix of codepage (unix charset)-encoded file names and UTF-8-encoded file names, the mess will take some effort to set straight.

A very helpful tool is available from Bjorn Jacke's convmv[1] work. Convmv is a tool that can be used to convert file and directory names from one encoding method to another. The most common use for this tool is to convert locale-encoded files to UTF-8 Unicode encoding.

8.1.1.6 Updates and Changes in Idealx smbldap-tools

The smbldap-tools have been maturing rapidly over the past year. With maturation comes change. The location of the `smbldap.conf` and the `smbldap_bind.conf` configuration files have been moved from the directory `/etc/smbldap-tools` to the new location of `/etc/opt/IDEALX/smblda-tools` directory.

The smbldap-tools maintains an entry in the LDAP directory in which it stores the next values that should be used for UID and GID allocation for POSIX accounts that are created using this tool. The DIT location of these values has changed recently. The original `sambaUnixIdPooldn object` entity was stored in a directory entry (DIT object) called `NextFreeUnixId`, this has been changed to the DIT object `sambaDomainName`. Anyone who updates from an older version to the current release should note that the information stored under `NextFreeUnixId` must now be relocated to the DIT object `sambaDomainName`.

8.2 Upgrading from Samba 1.x and 2.x to Samba-3

Sites that are being upgraded from Samba-2 (or earlier versions) to Samba-3 may experience little difficulty or may require a lot of effort, depending on the complexity of the configuration. Samba-1.9.x upgrades to Samba-3 will generally be simple and straightforward, although no upgrade should be attempted without proper planning and preparation.

There are two basic modes of use of Samba versions prior to Samba-3. The first does not use LDAP, the other does. Samba-1.9.x did not provide LDAP support. Samba-2.x could be compiled with LDAP support.

8.2.1 Samba 1.9.x and 2.x Versions Without LDAP

Where it is necessary to upgrade an old Samba installation to Samba-3, the following procedure can be followed:

Upgrading from a Pre-Samba-3 Version

1. Stop Samba. This can be done using the appropriate system tool that is particular for each operating system or by executing the **kill** command on **smbd**, **nmbd**, and **winbindd**.
2. Find the location of the Samba `smb.conf` file and back it up to a safe location.
3. Find the location of the `smbpasswd` file and back it up to a safe location.
4. Find the location of the `secrets.tdb` file and back it up to a safe location.

[1] <http://j3e.de/linux/convmv/>

5. Find the location of the lock directory. This is the directory in which Samba stores all its tdb control files. The default location used by the Samba Team is in `/usr/local/samba/var/locks` directory, but on Linux systems the old location was under the `/var/cache/samba` directory. However, the Linux Standards Base specified location is now under the `/var/lib/samba` directory. Copy all the tdb files to a safe location.

6. It is now safe to upgrade the Samba installation. On Linux systems it is not necessary to remove the Samba RPMs because a simple upgrade installation will automatically remove the old files. On systems that do not support a reliable package management system it is advisable either to delete the Samba old installation or to move it out of the way by renaming the directories that contain the Samba binary files.

7. When the Samba upgrade has been installed, the first step that should be completed is to identify the new target locations for the control files. Follow the steps shown in Section 8.1.1.4 to locate the correct directories to which each control file must be moved.

8. Do not change the hostname.

9. Do not change the workgroup name.

10. Execute the **testparm** to validate the `smb.conf` file. This process will flag any parameters that are no longer supported. It will also flag configuration settings that may be in conflict. One solution that may be used to clean up and to update the `smb.conf` file involves renaming it to `smb.conf.master` and then executing the following:

    ```
    root#  cd /etc/samba
    root#  testparm -s smb.conf.master > smb.conf
    ```

    The resulting `smb.conf` file will be stripped of all comments and of all nonconforming configuration settings.

11. It is now safe to start Samba using the appropriate system tool. Alternately, it is possible to just execute **nmbd**, **smbd**, and **winbindd** for the command line while logged in as the root user.

8.2.2 Applicable to All Samba 2.x to Samba-3 Upgrades

Samba 2.x servers that were running as a domain controller (PDC) require changes to the configuration of the scripting interface tools that Samba uses to perform OS updates for users, groups, and trust accounts (machines and interdomain).

The following parameters are new to Samba-3 and should be correctly configured. Please refer to Chapter 3, "Secure Office Networking" through Chapter 6, "A Distributed 2000-User Network" in this book for examples of use of the new parameters shown here:

add group script
add machine script
add user to group script
delete group script
delete user from group script
passdb backend
set primary group script

The *add machine script* functionality was previously handled by the *add user script*, which in Samba-3 is used exclusively to add user accounts.

Where the *passdb backend* used is either `smbpasswd` (the default) or the new `tdbsam`, the system interface scripts are typically used. These involve use of OS tools such as **useradd**, **usermod**, **userdel**, **groupadd**, **groupmod**, **groupdel**, and so on.

Where the *passdb backend* makes use of an LDAP directory, it is necessary either to use the `smbldap-tools` provided by Idealx or to use an alternate toolset provided by a third party or else home-crafted to manage the LDAP directory accounts.

8.2.3 Samba-2.x with LDAP Support

Samba version 2.x could be compiled for use either with or without LDAP. The LDAP control settings in the `smb.conf` file in this old version are completely different (and less complete) than they are with Samba-3. This means that after migrating the control files, it is necessary to reconfigure the LDAP settings entirely.

Follow the procedure outlined in Section 8.2.1 to affect a migration of all files to the correct locations.

The Samba SAM schema required for Samba-3 is significantly different from that used with Samba 2.x. This means that the LDAP directory must be updated using the procedure outlined in the Samba WHATSNEW.txt file that accompanies all releases of Samba-3. This information is repeated here directly from this file:

```
This is an extract from the Samba-3.0.x WHATSNEW.txt file:
==========================================================
Changes in Behavior
-------------------

The following issues are known changes in behavior between Samba 2.2 and
Samba 3.0 that may affect certain installations of Samba.

  1)  When operating as a member of a Windows domain, Samba 2.2 would
      map any users authenticated by the remote DC to the 'guest account'
      if a uid could not be obtained via the getpwnam() call.  Samba 3.0
      rejects the connection as NT_STATUS_LOGON_FAILURE.  There is no
      current work around to re-establish the 2.2 behavior.
```

```
  2)  When adding machines to a Samba 2.2 controlled domain, the
      'add user script' was used to create the UNIX identity of the
      machine trust account.  Samba 3.0 introduces a new 'add machine
      script' that must be specified for this purpose.  Samba 3.0 will
      not fall back to using the 'add user script' in the absence of
      an 'add machine script'

######################################################################
Passdb Backends and Authentication
##################################

There have been a few new changes that Samba administrators should be
aware of when moving to Samba 3.0.

  1) encrypted passwords have been enabled by default in order to
     inter-operate better with out-of-the-box Windows client
     installations.  This does mean that either (a) a samba account
     must be created for each user, or (b) 'encrypt passwords = no'
     must be explicitly defined in smb.conf.

  2) Inclusion of new 'security = ads' option for integration
     with an Active Directory domain using the native Windows
     Kerberos 5 and LDAP protocols.

     MIT kerberos 1.3.1 supports the ARCFOUR-HMAC-MD5 encryption
     type which is necessary for servers on which the
     administrator password has not been changed, or kerberos-enabled
     SMB connections to servers that require Kerberos SMB signing.
     Besides this one difference, either MIT or Heimdal Kerberos
     distributions are usable by Samba 3.0.


Samba 3.0 also includes the possibility of setting up chains
of authentication methods (auth methods) and account storage
backends (passdb backend).  Please refer to the smb.conf(5)
man page for details.  While both parameters assume sane default
values, it is likely that you will need to understand what the
values actually mean in order to ensure Samba operates correctly.

The recommended passdb backends at this time are

  * smbpasswd - 2.2 compatible flat file format
  * tdbsam - attribute rich database intended as an smbpasswd
    replacement for stand alone servers
  * ldapsam - attribute rich account storage and retrieval
    backend utilizing an LDAP directory.
  * ldapsam_compat - a 2.2 backward compatible LDAP account
```

```
    backend

Certain functions of the smbpasswd(8) tool have been split between the
new smbpasswd(8) utility, the net(8) tool, and the new pdbedit(8)
utility.  See the respective man pages for details.

################################################################################
LDAP
####

This section outlines the new features affecting Samba / LDAP
integration.

New Schema
----------

A new object class (sambaSamAccount) has been introduced to replace
the old sambaAccount.  This change aids us in the renaming of
attributes to prevent clashes with attributes from other vendors.
There is a conversion script (examples/LDAP/convertSambaAccount) to
modify and LDIF file to the new schema.

Example:

  $ ldapsearch .... -b "ou=people,dc=..." > sambaAcct.ldif
  $ convertSambaAccount --sid=<Domain SID> \
    --input=sambaAcct.ldif --output=sambaSamAcct.ldif \
    --changetype=[modify|add]

The <DOM SID> can be obtained by running 'net getlocalsid
<DOMAINNAME>' on the Samba PDC as root.  The changetype determines
the format of the generated LDIF output--either create new entries
or modify existing entries.

The old sambaAccount schema may still be used by specifying the
"ldapsam_compat" passdb backend.  However, the sambaAccount and
associated attributes have been moved to the historical section of
the schema file and must be uncommented before use if needed.
The 2.2 object class declaration for a sambaAccount has not changed
in the 3.0 samba.schema file.

Other new object classes and their uses include:

  * sambaDomain - domain information used to allocate rids
    for users and groups as necessary.  The attributes are added
    in 'ldap suffix' directory entry automatically if
    an idmap uid/gid range has been set and the 'ldapsam'
```

```
    passdb backend has been selected.

  * sambaGroupMapping - an object representing the
    relationship between a posixGroup and a Windows
    group/SID.  These entries are stored in the 'ldap
    group suffix' and managed by the 'net groupmap' command.

  * sambaUnixIdPool - created in the 'ldap idmap suffix' entry
    automatically and contains the next available 'idmap uid' and
    'idmap gid'

  * sambaIdmapEntry - object storing a mapping between a
    SID and a UNIX uid/gid.  These objects are created by the
    idmap_ldap module as needed.

  * sambaSidEntry - object representing a SID alone, as a Structural
    class on which to build the sambaIdmapEntry.


New Suffix for Searching
------------------------

The following new smb.conf parameters have been added to aid in directing
certain LDAP queries when 'passdb backend = ldapsam://...' has been
specified.

  * ldap suffix         - used to search for user and computer accounts
  * ldap user suffix    - used to store user accounts
  * ldap machine suffix - used to store machine trust accounts
  * ldap group suffix   - location of posixGroup/sambaGroupMapping entries
  * ldap idmap suffix   - location of sambaIdmapEntry objects

If an 'ldap suffix' is defined, it will be appended to all of the
remaining sub-suffix parameters.  In this case, the order of the suffix
listings in smb.conf is important.  Always place the 'ldap suffix' first
in the list.

Due to a limitation in Samba's smb.conf parsing, you should not surround
the DN's with quotation marks.
```

8.3 Updating a Samba-3 Installation

The key concern in this section is to deal with the changes that have been affected in Samba-3 between the Samba-3.0.0 release and the current update. Network administrators have expressed concerns over the steps that should be taken to update Samba-3 versions.

The information in Section 8.1.1.4 would not be necessary if every person who has ever produced Samba executable (binary) files could agree on the preferred location of the `smb.conf` file and other Samba control files. Clearly, such agreement is further away than a pipedream.

Vendors and packagers who produce Samba binary installable packages do not, as a rule, use the default paths used by the Samba-Team for the location of the binary files, the `smb.conf` file, and the Samba control files (tdb's as well as files such as `secrets.tdb`). This means that the network or UNIX administrator who sets out to build the Samba executable files from the Samba tarball must take particular care. Failure to take care will result in both the original vendor's version of Samba remaining installed and the new version being installed in the default location used by the Samba-Team. This can lead to confusion and to much lost time as the uninformed administrator deals with apparent failure of the update to take effect.

The best advice for those lacking in code compilation experience is to use only vendor (or Samba-Team) provided binary packages. The Samba packages that are provided by the Samba-Team are generally built to use file paths that are compatible with the original OS vendor's practices.

If you are not sure whether a binary package complies with the OS vendor's practices, it is better to ask the package maintainer via email than to waste much time dealing with the nuances. Alternately, just diagnose the paths specified by the binary files following the procedure outlined above.

8.3.1 Samba-3 to Samba-3 Updates on the Same Server

The guidance in this section deals with updates to an existing Samba-3 server installation.

8.3.1.1 Updating from Samba Versions Earlier than 3.0.5

With the provision that the binary Samba-3 package has been built with the same path and feature settings as the existing Samba-3 package that is being updated, an update of Samba-3 versions 3.0.0 through 3.0.4 can be updated to 3.0.5 without loss of functionality and without need to change either the `smb.conf` file or, where used, the LDAP schema.

8.3.1.2 Updating from Samba Versions between 3.0.6 and 3.0.10

When updating versions of Samba-3 prior to 3.0.6 to 3.0.6 through 3.0.10, it is necessary only to update the LDAP schema (where LDAP is used). Always use the LDAP schema file that is shipped with the latest Samba-3 update.

Samba-3.0.6 introduced the ability to remember the last *n* number of passwords a user has used. This information will work only with the `tdbsam` and `ldapsam` *`passdb backend`* facilities.

After updating the LDAP schema, do not forget to re-index the LDAP database.

8.3.1.3 Updating from Samba Versions after 3.0.6 to a Current Release

Samba-3.0.8 introduced changes in how the *username map* behaves. It also included a change in behavior of **winbindd**. Please refer to the man page for `smb.conf` before implementing any update from versions prior to 3.0.8 to a current version.

In Samba-3.0.11 a new privileges interface was implemented. Please refer to Section 5.3.1.1 for information regarding this new feature. It is not necessary to implement the privileges interface, but it is one that has been requested for several years and thus may be of interest at your site.

In Samba-3.0.11 there were some functional changes to the *ldap user suffix* and to the *ldap machine suffix* behaviors. The following information has been extracted from the WHATSNEW.txt file from this release:

```
============
LDAP Changes
============

If "ldap user suffix" or "ldap machine suffix" are defined in
smb.conf, all user-accounts must reside below the user suffix,
and all machine and inter-domain trust-accounts must be located
below the machine suffix.  Previous Samba releases would fall
back to searching the 'ldap suffix' in some cases.
```

8.3.2 Migrating Samba-3 to a New Server

The two most likely candidates for replacement of a server are domain member servers and domain controllers. Each needs to be handled slightly differently.

8.3.2.1 Replacing a Domain Member Server

Replacement of a domain member server should be done using the same procedure as outlined in Chapter 7, "Adding Domain Member Servers and Clients".

Usually the new server will be introduced with a temporary name. After the old server data has been migrated to the new server, it is customary that the new server be renamed to that of the old server. This will change its SID and will necessitate rejoining to the domain.

Following a change of hostname (NetBIOS name) it is a good idea on all servers to shut down the Samba **smbd**, **nmbd**, and **winbindd** services, delete the `wins.dat` and `browse.dat` files, then restart Samba. This will ensure that the old name and IP address information is no longer able to interfere with name to IP address resolution. If this is not done, there can be temporary name resolution problems. These problems usually clear within 45 minutes of a name change, but can persist for a longer period of time.

If the old domain member server had local accounts, it is necessary to create on the new domain member server the same accounts with the same UID and GID for each account. Where the *passdb backend* database is stored in the `smbpasswd` or in the `tdbsam` format, the user and group account information for UNIX accounts that match the Samba accounts will reside in the system `/etc/passwd`, `/etc/shadow`, and `/etc/group` files. In this case, be sure to copy these account entries to the new target server.

Where the user accounts for both UNIX and Samba are stored in LDAP, the new target server must be configured to use the **nss_ldap** tool set. This will automatically ensure that the appropriate user entities are available on the new server.

8.3.2.2 Replacing a Domain Controller

In the past, people who replaced a Windows NT4 domain controller typically installed a new server, created printers and file shares on it, then migrate across all data that was destined to reside on it. The same can of course be done with Samba.

From recent mailing list postings it would seem that some administrators have the intent to just replace the old Samba server with a new one with the same name as the old one. In this case, simply follow the same process as for upgrading a Samba 2.x system and do the following:

- Where UNIX (POSIX) user and group accounts are stored in the system `/etc/passwd`, `/etc/shadow`, and `/etc/group` files, be sure to add the same accounts with identical UID and GID values for each user.

  Where LDAP is used, if the new system is intended to be the LDAP server, migrate it across by configuring the LDAP server (`/etc/openldap/slapd.conf`). The directory can be populated either initially by setting this LDAP server up as a slave or by dumping the data from the old LDAP server using the **slapcat** command and then reloading the same data into the new LDAP server using the **slapadd** command. Do not forget to install and configure the **nss_ldap** tool and the `/etc/nsswitch.conf` (as shown in Chapter 5, "Making Happy Users").

- Copy the `smb.conf` file from the old server to the new server into the correct location as indicated previously in this chapter.

- Copy the `secrets.tdb` file, the `smbpasswd` file (if it is used), the `/etc/samba/passdb.tdb` file (only used by the `tdbsam` backend), and all the tdb control files from the old system to the correct location on the new system.

- Before starting the Samba daemons, verify that the hostname of the new server is identical to that of the old one. Note: The IP address can be different from that of the old server.

- Copy all files from the old server to the new server, taking precaution to preserve all file ownership and permissions as well as any POSIX ACLs that may have been created on the old server.

When replacing a Samba domain controller (PDC or BDC) that uses LDAP, the new server need simply be configured to use the LDAP directory, and for the rest it should just work.

The domain SID is obtained from the LDAP directory as part of the first connect to the LDAP directory server.

All Samba servers, other than one that uses LDAP, depend on the tdb files, and particularly on the `secrets.tdb` file. So long as the tdb files are all in place, the `smb.conf` file is preserved, and either the hostname is identical or the *`netbios name`* is set to the original server name, Samba should correctly pick up the original SID and preserve all other settings. It is sound advice to validate this before turning the system over to users.

8.3.3 Migration of Samba Accounts to Active Directory

Yes, it works. The Windows ADMT tool can be used to migrate Samba accounts to MS Active Directory. There are a few pitfalls to be aware of:

Migration to Active Directory

1. Administrator password must be THE SAME on the Samba server, the 2003 ADS, and the local Administrator account on the workstations. Perhaps this goes without saying, but there needs to be an account called `Administrator` in your Samba domain, with full administrative (root) rights to that domain.
2. In the Advanced/DNS section of the TCP/IP settings on your Windows workstations, make sure the *`DNS suffix for this connection`* field is blank.
3. Because you are migrating from Samba, user passwords cannot be migrated. You'll have to reset everyone's passwords. (If you were migrating from NT4 to ADS, you could migrate passwords as well.) To date this has not been attempted with roaming profile support; it has been documented as working with local profiles.
4. Disable the Windows Firewall on all workstations. Otherwise, workstations won't be migrated to the new domain.
5. When migrating machines, always test first (using ADMT's test mode) and satisfy all errors before committing the migration. Note that the test will always fail, because the machine will not have been actually migrated. You'll need to interpret the errors to know whether the failure was due to a problem or simply to the fact that it was just a test.

There are some significant benefits of using the ADMT, besides just migrating user accounts. ADMT can be found on the Windows 2003 CD.

- You can migrate workstations remotely. You can specify that SIDs be simply added instead of replaced, giving you the option of joining a workstation back to the old domain if something goes awry. The workstations will be joined to the new domain.
- Not only are user accounts migrated from the old domain to the new domain, but ACLs on the workstations are migrated as well. Like SIDs, ACLs can be added instead of replaced.
- Locally stored user profiles on workstations are migrated as well, presenting almost no disruption to the user. Saved passwords will be lost, just as when you administratively reset the password in Windows ADS.

- The ADMT lets you test all operations before actually performing the migration. Accounts and workstations can be migrated individually or in batches. User accounts can be safely migrated all at once (since no changes are made on the original domain). It is recommended to migrate only one or two workstations as a test before committing them all.