

Linux/Unix Active Directory Authentication Integration Using Samba Winbind

March 8, 2006

Prepared By:
Edwin Gnichtel

Table of Contents

INTRODUCTION	3
HOW WINBIND WORKS	4
Name Service Switch (NSS)	4
Pluggable Authentication Modules (PAM)	5
Kerberos	6
User and Group Mappings (IDMAP).....	6
Winbind IDMAP Options	6
Local TDB Database	6
OpenLDAP as an IDMAP Backend.....	7
Active Directory Application Partition as an IDMAP Backend.....	7
CONFIGURING WINBIND SUPPORTING COMPONENTS	8
Configuring Kerberos	8
Configuring the Name Service Switch	9
Configuring the PAM Subsystem.....	9
CONFIGURING WINBIND	12
Samba Winbind Standard (local IDMAP) Configuration.....	12
Winbind using LDAP IDMAP Back-End.....	13
Configuring OpenLDAP to support Winbind IDMAP.....	13
Winbind IDMAP LDAP configuration.....	15
Winbind using an Active Directory Application Partition for IDMAP Back-End	17
Configuring Windows Server 2003 Active Directory to support Winbind IDMAP	17
Winbind IDMAP LDAP configuration for AD Application Partitions.....	21
APPENDIX A. ADDITIONAL RESOURCES	23
APPENDIX B. ACTIVE DIRECTORY SCHEMA EXTENSIONS.....	24
APPENDIX C. ACKNOWLEDGEMENTS.....	39

Introduction

Samba is a set of Unix/Linux native components designed to provide Microsoft Windows authentication, file and print services. Samba, combined with its sub components, allows a Unix/Linux system to act much like a Windows Server, including the ability to natively join a Windows Active Directory domain, provide legacy NT 4 style domain services and fully integrated SMB file and printer sharing.

This document provides information for configuring the Winbind component of Samba, including the necessary supporting subcomponents used by Winbind. Winbind is a Unix/Linux daemon (service) that provides unified authentication against Windows Server security providers (Domain Controllers). Winbind can be configured to authenticate against either legacy Windows NT SAM domains or against Microsoft Windows Server Active Directory. This document will address AD based authentication only.

How Winbind Works

Winbind provides a Unix native view of Active Directory domain accounts and groups through a set of modules that leverage standard Unix components. Specifically, Winbind Name Services Switch (NSS) support and Pluggable Authentication Modules are used to pass requests to the Winbind Daemon which in turn queries the Active Directory (and supporting services, such as LDAP and Kerberos) to obtain the requested information, parse the information and reply to the calling components with the appropriate data.

As Winbind interacts with several layers of Unix type authentication subsystem components it is necessary to briefly discuss each of these components.

Name Service Switch (NSS)

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 an 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 server using MSRPC and LDAP calls (to AD) 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 an AD domain plus any trusted domains as though they were local Unix 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 that 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 specifies 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 an 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.

Pluggable Authentication Modules (PAM)

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 Server AD users into a UNIX system. This allows Windows Server AD users to log in to a UNIX machine and be authenticated against a suitable Domain Controller. These users can also change their passwords and have this change take effect directly on the 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.

Kerberos

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos is the core authentication protocol used by all native Active Directory clients including Samba Winbind.

User and Group Mappings (IDMAP)

When a user or group is created under Windows NT/200x it is allocated a numerical relative identifier (RID). This is slightly different from 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 users and groups. If a Windows user is resolved for the first time, it is allocated the next UNIX ID from the range. The same process applies for Windows groups. Over time, Winbind will have mapped all Windows users and groups, in a given domain or forest, to UNIX user IDs and group IDs.

The results of this mapping are stored persistently in an ID mapping database held in either a locally (host) stored TDB engine database or in a specified LDAP backend. This ensures that RIDs are mapped to UNIX IDs in a consistent way.

Winbind IDMAP Options

There are several options for configuring IDMAP information storage; local TDB Database, OpenLDAP based Idap store or Windows Server 2003 AD Application Partition (additional IDMAP and Idap options exist, but are not covered in this document).

Local TDB Database

The easiest IDMAP configuration is to use the default TDB database, which is locally stored on the host running Winbind. This configuration, however, has one significant draw-back: each host running Winbind will store IDMAP information in a unique sequence, preventing common UID and GID mappings between multiple hosts running Winbind.

This option is the preferred solution for large numbers of client Unix/Linux hosts that will never need common UID/GID to AD RID mappings. This solution works well for desktop Linux hosts that simply need access to AD for authentication purposes and will not be interacting with NFS based file sharing or will not be presenting Samba file and print shares.

OpenLDAP as an IDMAP Backend

OpenLDAP (slapd) is a GNU Open-Source LDAP version 3 compliant stand-alone directory service. Winbind can be configured to store IDMAP data in any LDAP V3 compliant directory server, however OpenLDAP is the most supported solution from an official Samba project standpoint.

If Winbind is to be used in conjunction with other Unix services, such as NFS, or common UID/GID to Windows AD RID mappings are desirable or necessary, it is recommended that an OpenLDAP server be configured to host IDMAP data.

Active Directory Application Partition as an IDMAP Backend

Windows Server 2003 Active Directory supports independent, manually configurable naming contexts, also referred to as Application Partitions. An application directory partition can contain a hierarchy of any type of objects, except security principals (AD Users, Groups and Computer account objects), and can be configured to replicate to any set of domain controllers in the forest. Unlike a domain partition (AD Domain), an application directory partition is not required to replicate to all domain controllers in a domain and the partition can replicate to domain controllers in different domains in the same forest.

By extending the AD Schema with the necessary NIS/Posix Account and Samba schema extensions, and creating an AD application partition, it is possible to unobtrusively store Winbind IDMAP data entries in the AD using one or more Domain controllers as IDMAP Idap backend servers. It is also possible to replicate this information in a simple and controlled manner to a subset of AD Domain Controllers located within either the same domain or in different domains in the same forest.

Note: Currently, AD Application Partition support as an IDMAP backend for Samba Winbind is still **experimental** and **should not be implemented as a production solution without careful environment specific testing**.

For more information on AD Application Partitions please see:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ad/ad/about_application_directory_partitions.asp.

Configuring Winbind Supporting Components

Most modern distributions of Unix type operating systems have pre-compiled packages for installing the core Samba components. It is necessary to verify that the latest installed versions of MIT Kerberos, PAM and the latest supporting libraries for NSS are installed on the host. Installation of Samba and its supporting components is beyond the scope of this document due to the nuances of each supported platform. However, due to the popularity of Samba, installation documentation for the various flavors of Unix and Linux are well documented and most current releases of Unix / Linux have Samba, and its supporting components, as available options during installation (Appendix A. contains a list of sites for further information on each of the mentioned components).

The following sections describe how to configure Samba Winbind and supporting components.

Configuring Kerberos

To configure Kerberos it is necessary to edit the “krb5.conf” configuration file. This file is usually located in the “/etc” directory. Edit the file as shown below (figure 1.) to include your AD fully qualified domain name (FQDN) as the “realm” under the “libdefaults” section and make sure to set DNS lookups for realm and KDC to true as this will allow DNS to be used to resolve both services. Also edit the “domain realm” section, replacing the place holder entries with the AD domain name mapping (the first entry with a “period” preceding the domain name is not a typo).

Note: An AD domain is also a Kerberos realm and the names are used interchangeably in this document.

```
[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log

[libdefaults]
default_realm = xxx.addomainname.com
dns_lookup_realm = true
dns_lookup_kdc = true

[realms]
EXAMPLE.COM = {
kdc = kerberos.example.com:88
admin_server = kerberos.example.com:749
default_domain = example.com
}

[domain_realm]
.xxx.domainname.com = XXX.DOMAINNAME.COM
```

```
xxx.domainname.com = XXX.DOMAINNAME.COM

[kdc]
profile = /var/kerberos/krb5kdc/kdc.conf

[appdefaults]
pam = {
debug = false
ticket_lifetime = 36000
renew_lifetime = 36000
forwardable = true
krb4_convert = false
}
```

Figure 1. Kerberos configuration file.

Configuring the Name Service Switch

The name service switch (NSS) configuration is held in the “nsswitch.conf” configuration file, which is normally located in the “/etc” directory. The following Winbind entries, as shown in figure 2, should be present to instruct the NSS subsystem to use Winbind for the necessary name related services.

```
#
# /etc/nsswitch.conf
#
passwd:      files winbind
shadow:     files winbind
group:      files winbind
hosts:      files dns
bootparams: nisplus [NOTFOUND=return] files
ethers:     files
netmasks:  files
networks:   files
protocols:  files winbind
rpc:        files
services:   files winbind
netgroup:   files winbind
publickey:  nisplus
automount:  files winbind
aliases:    files nisplus
```

Figure 2. Name service switch configuration file.

Configuring the PAM Subsystem

Depending on the flavor Unix or Linux being configured, PAM configuration differs slightly. This is due to certain configurations using the pam-stack (“pam_stack.so”) configuration.

Traditionally, each service that uses PAM has its own PAM configuration. For example, the “login” service (text mode TTY login) has a file named “login” located in the “/etc/pam.d” directory. Inside this file would be a series of entries defining the PAM modules to be used for the various authentication steps needed by the login service.

However, if a system is setup to use a stackable configuration, the “pam_stack.so” module might be used to allow the login service to call outside its own “stack”, or configuration file, to other services’ PAM stacks or a central file. When using a centrally configured PAM implementation, the single file commonly used for configuring all services stacks is the “system-auth” file located in “etc/pam.d”

The following example PAM stack configuration is valid for either a single service or for systems using “pam_stack.so” and a “system-auth” stack configuration file. If it is noted that a service specific PAM configuration file, such as “/etc/pam.d/login” contains a “pam_stack.so service=system-auth” entry (or similar pam_stack.so entry), then the “system-auth” file should be the only file modified to include the necessary Winbind entries.

Figure 3 illustrates a working example of a PAM stack configuration file. It should be possible to cut and paste this directly into the “system-auth” PAM stack configuration file.

```

auth    required    /lib/security/$ISA/pam_env.so
auth    sufficient  /lib/security/$ISA/pam_unix.so likeauth nullok
auth    sufficient  /lib/security/$ISA/pam_krb5.so use_first_pass
auth    sufficient  /lib/security/$ISA/pam_winbind.so use_first_pass
auth    required    /lib/security/$ISA/pam_deny.so

account sufficient  /lib/security/$ISA/pam_succeed_if.so uid < 100
account required    /lib/security/$ISA/pam_unix.so
account sufficient  /lib/security/$ISA/pam_krb5.so
account sufficient  /lib/security/$ISA/pam_winbind.so

password requisite  /lib/security/$ISA/pam_cracklib.so retry=3
password sufficient /lib/security/$ISA/pam_unix.so nullok use_authok md5 shadow
password sufficient /lib/security/$ISA/pam_krb5.so use_authok
password sufficient /lib/security/$ISA/pam_winbind.so use_authok
password required   /lib/security/$ISA/pam_deny.so

session required    /lib/security/$ISA/pam_limits.so
session required    /lib/security/$ISA/pam_unix.so
session optional    /lib/security/$ISA/pam_mkhomedir.so skel=etc/skel/ umask=0022
session optional    /lib/security/$ISA/pam_krb5.so

```

Figure 3 Example PAM Configuration file (note: some lines are wordwrapped)

The Kerberos and Winbind specific entries that need to be added to a PAM configuration file for Winbind are as follows:

```
auth    sufficient /lib/security/$ISA/pam_krb5.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_winbind.so use_first_pass

account sufficient /lib/security/$ISA/pam_krb5.so
account sufficient /lib/security/$ISA/pam_winbind.so

password sufficient /lib/security/$ISA/pam_krb5.so use_authtok
password sufficient /lib/security/$ISA/pam_winbind.so use_authtok

session optional /lib/security/$ISA/pam_mkhomedir.so skel=etc/skel/ umask=0022
session optional /lib/security/$ISA/pam_krb5.so
```

Note: For automatic home directory creation, it is necessary to have the “pam_mkhomedir.so” module present with “skel=etc/skel/ umask=0022” parameters set. If a user authenticates and no home directory exists, the home directory is created in /home. The umask=0022 parameter causes the directory permission to be set to 755. Use the umask permissions mask flag to strengthen or weaken permissions as needed (this follows standard umask convention). The home directory is constructed from the skeletal files found in the /etc/skel directory. It may be necessary to install “pam_mkhomedir.so” if the system being configured is an older distribution of Linux or Unix (see Appendix A. for reference locations).

Configuring Winbind

The following sections describe the necessary global settings for the “smb.conf” configuration file located in “etc/samba”. The sections are separated only by the IDMAP configuration options and each example demonstrates a working configuration.

Samba Winbind Standard (local IDMAP) Configuration

The following smb.conf configuration file, shown in figure 4, is configured to support a local, non-centralized, Winbind IDMAP database. Only the “global” section is included as this is the only section required for Winbind configuration. For additional Samba “smb.conf” reference the support sites listed in Appendix A.

```
[global]
workgroup = NETBIOSDOMAINNAME
netbios name = NETBIOSCOMPNAME
server string = A Workstation
printcap name = /etc/printcap
load printers = yes
log file = /var/log/samba/%m.log
max log size = 50
security = ADS
realm = XXX.DOMAINNAME.COM
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
allow trusted domains = yes
unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\n
*passwd:*all*authentication*tokens*updated*su\
ccessfully*
pam password change = yes
obey pam restrictions = yes
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
dns proxy = no
idmap uid = 10000-10000000
idmap gid = 10000-10000000
winbind use default domain = yes
winbind separator = -
winbind enum users = yes
winbind enum groups = yes
template shell = /bin/bash
template homedir = /home/%U
# %U=username
```

Figure 4 Example simple Samba configuration file “Global” section (note: some lines are wordwrapped)

For the “workgroup” entry, enter the legacy NetBIOS name of the AD domain the host will be participating in. For the “netbiosname” entry, set this value to the computer name

the host should be known by in AD (e.g. the computer account name). For the “realm” entry, set the value to the FQDN of the Active Directory domain this host will be joined to.

Note: In an AD domain containing large numbers of user and group objects (over 3000), it is recommended that the “winbind enum users” and “winbind enum groups” option be set to a “no” value. While this will impact certain user interface elements from presenting all available AD security principals, it may dramatically improve system response time for certain actions and will significantly reduce LDAP traffic.

Once Winbind and the supporting components are configured, issue the following command at a shell (logged in as root) to finalize the Winbind setup:

```
“net ads join -U administrator”
```

This command will join the host to the AD domain (supplement “administrator” with any account granted the necessary rights to join computers to the AD).

To start and test Winbind, type “winbindd” at the shell prompt (as root). To validate that Winbind is correctly seeing the domain, type “wbinfo -g” at the shell prompt; this should enumerate and print to screen all the groups in the AD domain (for further “wbinfo” options, type “wbinfo” at the shell). To validate authentication is functioning correctly, test logons through several different services including “ssh” and local login.

Once testing is complete, and the configuration has been validated as working, add the Winbind daemon (winbindd) to the appropriate Unix/Linux run-level configuration (consult the OS specific documentation for assistance with this step).

Winbind using LDAP IDMAP Back-End

In order to configure Winbind to use an LDAP directory server, it is necessary to configure an LDAP back-end. This document specifically references using OpenLDAP as the LDAP provider for this configuration; however other stand-alone LDAP v3 compliant directory services will also suffice but are not within the scope of this document.

Configuring OpenLDAP to support Winbind IDMAP

Installation of OpenLDAP is not covered in this document; please reference the official OpenLDAP documentation located at <http://www.openldap.org>. Once OpenLDAP is installed, configure the “slapd.conf”, located in “/etc/slapd” as shown in figure 5. To generate the “rootdn” password for OpenLDAP, use the “slappasswd” command to generate a password hash, then paste the hash into the “rootpw” value. Be sure to validate the file is set to mode 700 for permissions.

Note: As with all example configurations, replace placeholder name references, such as “dc=domainname,dc=com”, with appropriate names. In the case of OpenLDAP, this

name need not match the Active Directory naming conventions as the two are entirely independent.

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#

include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema
include      /etc/ldap/schema/openldap.schema
include      /etc/ldap/schema/samba.schema

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd/slapd.pid
# List of arguments that were passed to the server
argsfile     /var/run/slapd.args

# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn. (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!

#####
# BDB database definitions
#####

database     bdb
suffix       "dc=domainname,dc=com"
rootdn       "cn=Manager,dc=domainname,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw       {SSHA}YoFP07ouw2h4xiVyMOJqQwfCq3w/OerG
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory    "C:/openldap/var/openldap-data"
# Indices to maintain
index objectClass      eq
###index uid            pres,eq
###index rid            eq
index uidNumber        eq
index gidNumber        eq
index cn                eq
index sambaSID          eq
```

Figure 6. Example OpenLDAP configuration file (note: some lines are wordwrapped)

Prior to starting the OpenLDAP server, validate the “samba.schema” schema file is located in the “etc/ldap/schema” directory. The latest version of the schema can be found in the samba distribution files available from www.samba.org.

Once the OpenLDAP configuration is complete, start the OpenLDAP server by typing “slapd” at the shell prompt. Validate that the LDAP server is functioning by using an LDAP tool, such as LDP, to connect and bind to the server (using the cn=Manager,dc=domainname,dc=com account and the password entered in slapd.conf). Create an object named “cn=Manager, dc=domainname, dc=com” with an object class type of “organizationalRole” and a description of “Directory Manager”.

The above example does not include SSL support. It is absolutely critical that once a working configuration is achieved and tested, that SSL be enabled for LDAP connections. This is to prevent the “rootdn” password from being compromised during a simple LDAP bind from the Winbind host. For additional information on enabling SSL for LDAP connections in OpenLDAP see:

http://www.openldap.org/pub/ksoper/OpenLDAP_TLS_howto.html#4.0

Once testing is complete, and the configuration has been validated as working, add the OpenLDAP daemon (slapd) to the appropriate Unix/Linux run-level configuration (consult the OS specific documentation for assistance with this step).

Winbind IDMAP LDAP configuration

Configuring the smb.conf to instruct Winbind to use an LDAP backend for IDMAP data storage is fairly straight forward. Add the following lines to a working “smb.conf” configuration file:

```
ldap admin dn = cn=Manager,dc=domainname,dc=com
ldap idmap suffix = ou=idmap
ldap suffix = dc=domainname,dc=com
ldap backend = ldap:ldap://FQDNofLDAPSERVER
```

Figure 7. Samba configuration additions for IDMAP LDAP backend

The “ldap admin dn” value should match the rootdn value set during the configuration of the OpenLDAP server. The “ldap idmap suffix” should be set to “ou=idmap”. The ldap suffix should match the “suffix” value set in the “slapd.conf” configuration file. The “ldap backend” value should be the IP address or fully qualified DNS name of the server running OpenLDAP.

Once the above settings are present, type the following command at the shell prompt:

```
“smbpassword -w [password_of_Directory_Manager]”
```

Where “password_of_Directory_Manager” would be the password set for the OpenLDAP rootdn account (use the human readable form submitted to the “slappasswd” command). The password will be stored in the Samba “secrets.ldb” database, which hashes the password and is only readable by the “root” account.

Once Winbind and the supporting components are configured, issue the following command at a shell (logged in as root) to finalize the Winbind setup:

```
“net ads join -U administrator”
```

This command will join the host to the AD domain (supplement “administrator” with any account granted the necessary rights to join computers to the AD).

To start and test Winbind, type “winbindd” at the shell prompt (as root). To validate that Winbind is correctly seeing the domain, type “wbinfo -g” at the shell prompt; this should enumerate and print to screen all the groups in the AD domain (for further “wbinfo” options, type “wbinfo” at the shell). To validate authentication is functioning correctly, test logons through several different services including “ssh” and local login.

To validate that IDMAP entries are being entered correctly in the LDAP directory, connect with an LDAP administration tool (e.g. LDP) to the OpenLDAP server. Expand the IDMAP container (ou=IDMAP); there should be numerous entries similar to the example below:

```
Expanding base 'sambaSID=S-1-5-21-1033264847-1678921569-1609722162-1411,ou=idmap,dc=domainname,dc=com'...
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn: sambaSID=S-1-5-21-1033264847-1678921569-1609722162-1411,ou=idmap,dc=domainname,dc=com
    2> objectClass: sambaldmapEntry; sambaSidEntry;
    1> gidNumber: 10218;
    1> sambaSID: S-1-5-21-1033264847-1678921569-1609722162-1411;
```

Figure 8. Example IDMAP entry

Once testing is complete, and the configuration has been validated as working, add the Winbind daemon (winbindd) to the appropriate Unix/Linux run-level configuration (consult the OS specific documentation for assistance with this step).

To configure Winbind to use LDAP SSL for connections, add the following line to the “smb.conf” configuration file:

```
ldap ssl = on
```

Also change the idmap backend value to include an “s” in the URL address as shown below:

```
idmap backend = ldap:ldaps://FQDNofLDAPSERVER
```

Winbind using an Active Directory Application Partition for IDMAP Back-End

Active Directory, as with any LDAP V3 compliant directory service, can function as the backend for Winbind IDMAP back-end storage. This configuration uses a Windows Server 2003 Active Directory application partition to store the IDMAP entries.

Using an application partition has some significant advantages over storing the IDMAP data in an AD domain partition or in the configuration partition: security principals can not be created in an application partition, preventing possible rogue accounts from being added using the “ldap admin dn” account info stored in the Samba “secrets.tdb” database. Additionally, the application partition replication scope is completely controllable (application partitions do not participate in global catalog indexing) allowing strict selection of one or more Domain Controllers to host the IDMAP data, which negates the risk of an LDAP flooding attack impacting domain or forest-wide function.

Due to the potential issues with not using an AD application partition, it is not recommended that Winbind IDMAP be configured for use with a Domain Partition or the Configuration Partition. As such, Windows 2000 Active Directory is not recommended for use with Winbind for IDMAP storage.

Configuring Windows Server 2003 Active Directory to support Winbind IDMAP

Note: The tasks outlined in this section assume that the actions affecting Active Directory are being executed by an account with Enterprise Admins and Schema Admins privileges.

Prior to configuring Winbind IDMAP to store data in the AD, it is necessary to extend the AD schema with the necessary Samba schema extensions. To perform this action, login with Schema Admins privileges on the Active Directory Forest Schema Master domain controller, unzip the “ADSambaSchema.zip” archive (included with this document package) to “c:\sambaschema”. Open each “.ldf” file and perform a search and replace on the string “dc=testnet,dc=com”, replacing the string with the top level Domain Component (DC=) values for the **AD forest**. Once the files have been modified correctly, install the extensions by executing the “schemaupdate.bat” batch file from the command prompt. Be sure to change directory to “c:\sambaschema” prior to executing the batch file.

Once the schema extensions have been loaded successfully, open the schema management MMC snap-in. Validate that the “uidNumber” and “gidNumber” attributes have no minimum or maximum value setting by viewing the properties of the attribute objects as shown in figure 9.

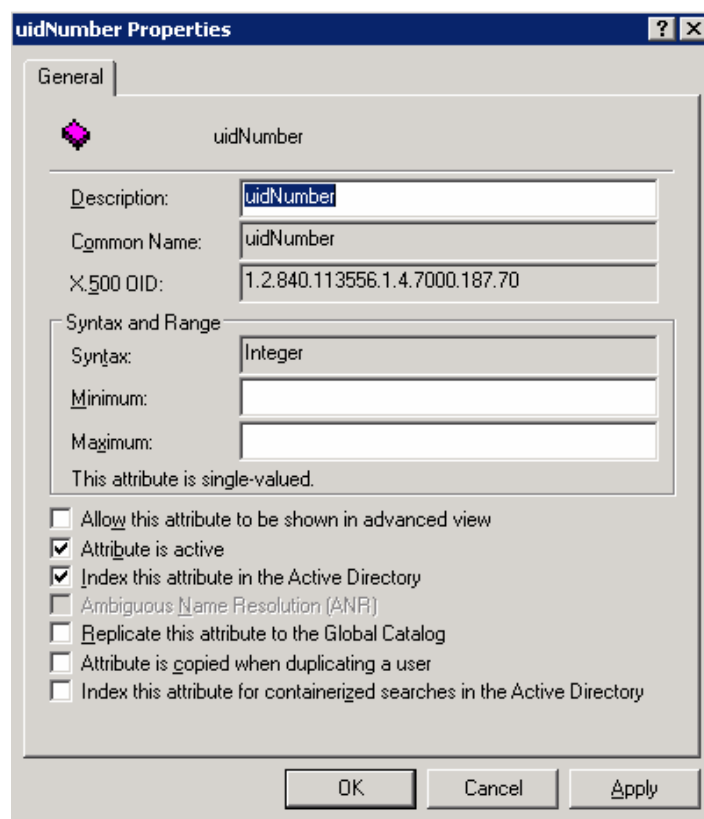


Figure 9. uidNumber attribute properties

Having completed the schema updates and verified the “uidNumber” and “gidNumber” attributes, it is necessary to create a new user account, or group, in the Active Directory. This group will be used to set access control on the application partition created in the next step. If a single account is created, this account must be located in the same domain that will house the DC holding the application partition, as Winbind uses a simple LDAP bind for authentication. Additionally, if a group is used, the group can be housed anywhere in the forest, but an account must be present in the local domain partition of any DC holding a replica of the partition and being used for Winbind IDMAP.

Note: For the purposes of this documentation, a single user account named “IDMAPManager” will be used to demonstrate the ACL settings for the application partition.

To create an application partition:

1. Open a Command Prompt on the DC that should hold the first replica of the application partition.
2. Type: **ntdsutil**
3. At the ntdsutil command prompt, type: **domain management**
4. At the domain management command prompt, type: **connection**

5. At the connection command prompt, type: **connect to server name of domain controller**
6. At the connection command prompt, type: **quit**
7. At the domain management command prompt, do type: **create nc dc=sambaidmap,dc=domainname,dc=com null**

Once the application partition has been created, open ADSIedit.msc and connect to the application partition using settings similar to those shown in figure 10.

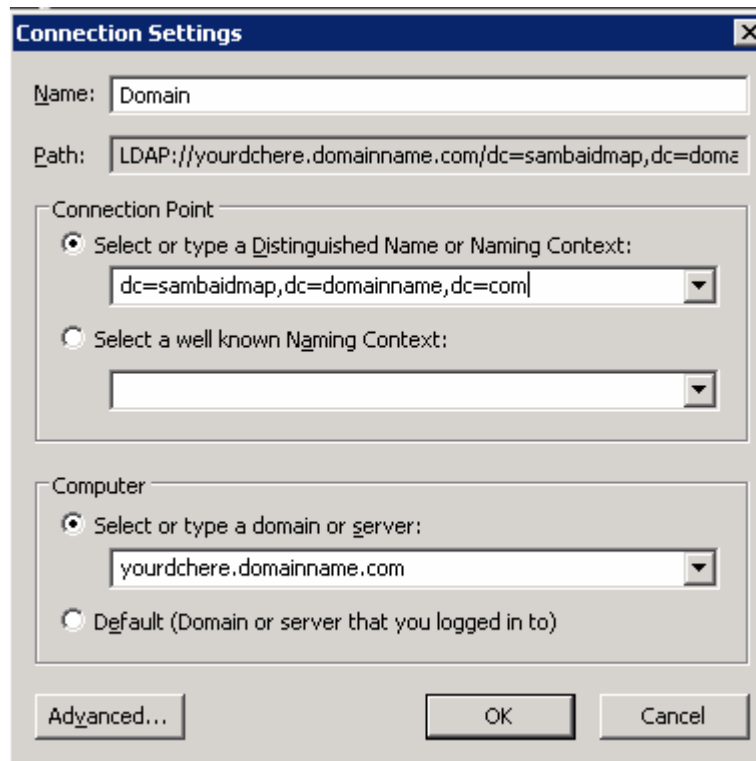


Figure 10. Example ADSI connection setting for accessing an application partition

Once connected, select the top level application partition (e.g. dc=sambaidmap,dc=domainname,dc=com) node in the left-hand pane and right-click selecting “new object” from the object list, select “sambaUnixIdPool”. When asked to enter the “ou=” attribute, type “idmap”. When queried for the mandatory “gidNumber” and “uidNumber” values, type in “10000” for each.

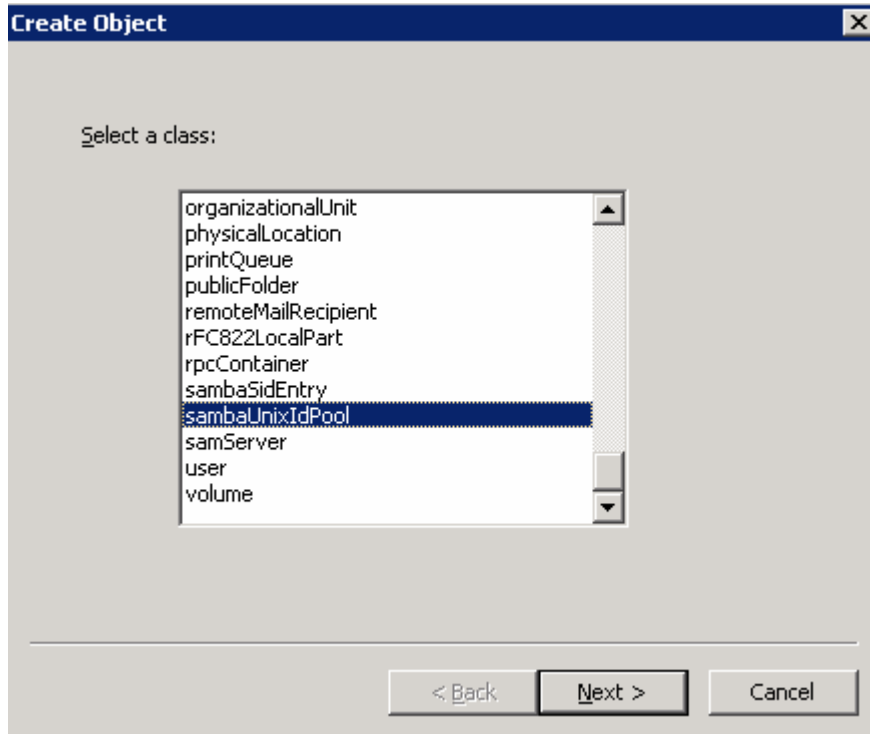


Figure 11. Create Object list

Once the “ou=idmap,dc=sambaidmap,dc=domainname,dc=com” container has been created, right-click on the object and select properties. On the security tab, click “Add” and proceed to add the “IDMAPManager” user account. Grant this account “Read,Write, Create All Child Objects, Delete All Child Objects” as shown in figure 12.

Permissions for idmap	Allow	Deny
Full Control	<input type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create All Child Objects	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delete All Child Objects	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Special Permissions	<input type="checkbox"/>	<input type="checkbox"/>

Figure 12. Permissions for the “IDMAPManager” account on the “ou=IDMAP” container

Having completed the steps above, the AD should now be ready to host IDMAP data. As with the OpenLDAP configuration, once the final configuration has been tested, SSL should be used to protect the “IDMAPManager” credentials. Any DC participating in an AD Forest with a Microsoft Enterprise CA implementation will already be capable of accepting SSL connections. To implement LDAP SSL without a Microsoft Enterprise CA, or for additional information configuring DC’s to use SSL, see:

<http://www.microsoft.com/technet/security/guidance/secmod154.mspx>

Winbind IDMAP LDAP configuration for AD Application Partitions

Configuring the smb.conf to instruct Winbind to use an LDAP backend for IDMAP data storage in an AD application partition is nearly identical to the configuration used for OpenLDAP. Add the following lines to a working “smb.conf” configuration file:

```
ldap admin dn = cn=IDMAPManager,cn=users,dc=domainname,dc=com
ldap idmap suffix = ou=idmap
ldap suffix = dc=sambaidmap,dc=domainname,dc=com
idmap backend = ldap:ldap://FQDNofADDomainController
```

Figure 13. Samba configuration additions for IDMAP AD backend

The “ldap admin dn” value should match the LDAP Distinguished Name (DN) of the “IDMAPManager” created in AD (or other user with rights to the idmap container). The “ldap idmap suffix” should be set to “ou=idmap”. The ldap suffix should match the DN of the application partition. The “idmap backend” value should be the IP address or fully qualified DNS name of the AD domain controller.

Once the above settings are present, type the following command at the shell prompt:

```
“smbpassword -w [password_of_IDMAPManager]”
```

Where “password_of_IDMAPManager” would be the password for the IDMAPManager account in AD. The password will be stored in the Samba “secrets.ldb” database, which hashes the password and is only readable by the “root” account.

Once Winbind and the supporting components are configured, issue the following command at a shell (logged in as root) to finalize the Winbind setup:

```
“net ads join -U administrator”
```

This command will join the host to the AD domain (supplement “administrator” with any account granted the necessary rights to join computers to the AD).

To start and test Winbind, type “winbindd” at the shell prompt (as root). To validate that Winbind is correctly seeing the domain, type “wbinfo -g” at the shell prompt; this should enumerate and print to screen all the groups in the AD domain (for further “wbinfo” options, type “wbinfo” at the shell). To validate authentication is functioning correctly, test logons through several different services including “ssh” and local login.

To validate that IDMAP entries are being entered correctly in the Active Directory application partition, connect with an LDAP administration tool (e.g. LDP) or ADSIEdit to the AD Application Partition. Expand the IDMAP container (ou=IDMAP); there should be numerous entries similar to the example below:

```

Expanding base 'sambaSID=S-1-5-21-1033264847-1678921569-1609722162-
1002,OU=idmap,DC=sambaidmap,DC=domainname,DC=com'...
Result <0>: (null)
Matched DNs:
Getting 1 entries:
>> Dn: sambaSID=S-1-5-21-1033264847-1678921569-1609722162-
1002,OU=idmap,DC=sambaidmap,DC=domainname,DC=com
    1> uidNumber: 10017;
    1> sambaSID: S-1-5-21-1033264847-1678921569-1609722162-1002;
    3> objectClass: top; sambaldmapEntry; sambaSidEntry;
    1> distinguishedName: sambaSID=S-1-5-21-1033264847-1678921569-
1609722162-1002,OU=idmap,DC=sambaidmap,DC=domainname,DC=com;
    1> instanceType: 0x4 = ( IT_WRITE );
    1> whenCreated: 12/12/2004 19:46:11 Eastern Standard Time Eastern Daylight
Time;
    1> whenChanged: 12/12/2004 19:46:11 Eastern Standard Time Eastern Daylight
Time;
    1> uSNCreated: 1364776;
    1> uSNChanged: 1364776;
    1> name: S-1-5-21-1033264847-1678921569-1609722162-1002;
    1> objectGUID: cb129204-8c39-4793-ac85-786ed858a65d;
    1> objectCategory:
CN=sambaSidEntry,CN=Schema,CN=Configuration,DC=domainname,DC=com;

```

Figure 8. Example IDMAP entry

Once testing is complete, and the configuration has been validated as working, add the Winbind daemon (winbindd) to the appropriate Unix/Linux run-level configuration (consult the OS specific documentation for assistance with this step).

To configure Winbind to use LDAP SSL for connections, add the following line to the “smb.conf” configuration file:

```
ldap ssl = on
```

Also change the idmap backend value to include an “s” in the URL address as shown below:

```
idmap backend = ldap:ldaps://FQDNofADDomainController
```

Appendix A. Additional Resources

Samba Information: <http://www.samba.org>

Kerberos Information: <http://web.mit.edu/kerberos/www/>

Kerberos Configuration Files:

<http://www.lns.cornell.edu/public/COMP/krb5/krb5-admin/Configuration-Files.html#Configuration%20Files>

Linux PAM Information: <http://www.kernel.org/pub/linux/libs/pam/>

Kerberos Information: <http://web.mit.edu/kerberos/www/>

Linux NSS Information:

http://www.gnu.org/software/libc/manual/html_node/Name-Service-Switch.html

SUN General Information: <http://docs.sun.com/app/docs>

SUN PAM Information: <http://www.sun.com/software/solaris/pam/>

SUN Samba Software Modules: <http://www.sun.com/software/solaris/pam/>

SUN pam_mkhome.so information: http://keutel.de/pam_mkhome/

OpenLDAP Information: <http://www.openldap.org>

OpenLDAP and SSL:

http://www.openldap.org/pub/ksoper/OpenLDAP_TLS_howto.html#4.0

Active Directory Application Partitions:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ad/ad/about_application_directory_partitions.asp

Active Directory and SSL:

<http://www.microsoft.com/technet/security/guidance/secmod154.msp>

Appendix B. Active Directory Schema Extensions

Posixattributes.ldf

```
dn: CN=uidNumber,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: uidNumber
attributID: 1.3.6.1.1.1.1.0
attributeSyntax: 2.5.5.9
cn: uidNumber
distinguishedName: CN=uidNumber,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: uidNumber
name: uidNumber
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
rangeLower: 2000
rangeUpper: 10000
schemaDGUID:: 58MKDRf3G0aAARxcPo+TNA==
showInAdvancedViewOnly: TRUE

dn: CN=gidNumber,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: gidNumber
attributID: 1.3.6.1.1.1.1.1
attributeSyntax: 2.5.5.9
cn: gidNumber
distinguishedName: CN=gidNumber,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: gidNumber
name: gidNumber
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
rangeLower: 2000
rangeUpper: 10000
schemaDGUID:: uBkKU2nqeUKB9Lg/ituBLw==
showInAdvancedViewOnly: TRUE

dn: CN=loginShell,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: loginShell
attributID: 1.3.6.1.1.1.1.4
attributeSyntax: 2.5.5.5
cn: loginShell
distinguishedName: CN=loginShell,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: loginShell
name: loginShell
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 22
schemaDGUID:: 2mImn2DEJ0e8vcTzTDQMLA==
showInAdvancedViewOnly: TRUE

dn: CN=gecos,CN=Schema,CN=Configuration,dc=testnet,dc=com
```

```
changetype: add
adminDisplayName: geccos
attributeID: 1.3.6.1.1.1.1.2
attributeSyntax: 2.5.5.5
cn: geccos
distinguishedName: CN=geccos,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: geccos
name: geccos
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 22
schemaIDGUID:: /qrVb1X+d0WdR3gMHhYn1w==
showInAdvancedViewOnly: TRUE

dn: CN=memberUid,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: memberUid
attributeID: 1.3.6.1.1.1.1.12
attributeSyntax: 2.5.5.5
cn: memberUid
distinguishedName: CN=memberUid,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: FALSE
LDAPDisplayName: memberUid
name: memberUid
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 22
schemaIDGUID:: 5u4QiSA4NUOKpGny9udGjQ==
showInAdvancedViewOnly: TRUE

dn: CN=shadowExpire,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowExpire
attributeID: 1.3.6.1.1.1.1.10
attributeSyntax: 2.5.5.9
cn: shadowExpire
distinguishedName: CN=shadowExpire,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowExpire
name: shadowExpire
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: o63EQPvXtUaiEF2lyPUEdw==
showInAdvancedViewOnly: TRUE

dn: CN=shadowFlag,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowFlag
attributeID: 1.3.6.1.1.1.1.11
attributeSyntax: 2.5.5.9
cn: shadowFlag
distinguishedName: CN=shadowFlag,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowFlag
name: shadowFlag
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
```

```
schemaIDGUID:: c1JtdrBiMUqiCIP7VGAZ0g==
showInAdvancedViewOnly: TRUE

dn: CN=shadowInactive,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowInactive
attributeID: 1.3.6.1.1.1.1.9
attributeSyntax: 2.5.5.9
cn: shadowInactive
distinguishedName: CN=shadowInactive,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowInactive
name: shadowInactive
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: eRWW0Mi8v0SKIECp/uQGzg==
showInAdvancedViewOnly: TRUE

dn: CN=shadowLastChange,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowLastChange
attributeID: 1.3.6.1.1.1.1.5
attributeSyntax: 2.5.5.9
cn: shadowLastChange
distinguishedName: CN=shadowLastChange,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowLastChange
name: shadowLastChange
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: xDf7Rhtk4keGQ1u3KqGkTw==
showInAdvancedViewOnly: TRUE

dn: CN=shadowMax,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowMax
attributeID: 1.3.6.1.1.1.1.7
attributeSyntax: 2.5.5.9
cn: shadowMax
distinguishedName: CN=shadowMax,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowMax
name: shadowMax
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: EFk3wWJoRUeBu/HS58T+Jg==
showInAdvancedViewOnly: TRUE

dn: CN=shadowMin,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowMin
attributeID: 1.3.6.1.1.1.1.6
attributeSyntax: 2.5.5.9
cn: shadowMin
distinguishedName: CN=shadowMin,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: shadowMin
```

```

name: shadowMin
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: Jndf3iFO0kmlknWk6qKvqw==
showInAdvancedViewOnly: TRUE

dn: CN=shadowWarning,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowWarning
attributeID: 1.3.6.1.1.1.1.8
attributeSyntax: 2.5.5.9
cn: shadowWarning
distinguishedName: CN=shadowWarning,CN=Schema,CN=Configuration,dc=testnet,dc=com
instanceType: 4
isSingleValued: TRUE
IDAPDisplayName: shadowWarning
name: shadowWarning
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: attributeSchema
oMSyntax: 2
schemaIDGUID:: S9nkW4wmIUKvFIPMsxz2IQ==
showInAdvancedViewOnly: TRUE

DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-

```

Posixclasses.ldf

```

dn: CN=posixAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: posixAccount
cn: posixAccount
defaultSecurityDescriptor:
D:(A;;LCRPLOCR;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=posixAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
governorID: 1.3.6.1.1.1.2.0
instanceType: 4
IDAPDisplayName: posixAccount
mayContain: description
mayContain: gecos
mayContain: loginShell
mayContain: userPassword
mustContain: cn
mustContain: gidNumber
mustContain: homeDirectory
mustContain: uid
mustContain: uidNumber
name: posixAccount
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
rDNAttID: cn
schemaIDGUID:: bJwa/uRYrEy0MmcBZzXYHQ==
showInAdvancedViewOnly: TRUE

```

```

subClassOf: top
systemOnly: FALSE

dn: CN=shadowAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: shadowAccount
cn: shadowAccount
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=shadowAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.1.1.2.1
instanceType: 4
IDAPDisplayName: shadowAccount
mayContain: description
mayContain: shadowExpire
mayContain: shadowFlag
mayContain: shadowInactive
mayContain: shadowLastChange
mayContain: shadowMax
mayContain: shadowMin
mayContain: shadowWarning
mayContain: userPassword
mustContain: uid
name: shadowAccount
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
rDNAttID: cn
schemalDGUID:: 03MjwPrR6ECUyHB/jvIYQA==
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE

dn: CN=posixGroup,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: posixGroup
cn: posixGroup
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=posixGroup,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.1.1.2.2
instanceType: 4
IDAPDisplayName: posixGroup
mayContain: description
mayContain: memberUid
mayContain: userPassword
mustContain: cn
mustContain: gidNumber
name: posixGroup
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: group
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
rDNAttID: cn
schemalDGUID:: 7pmYcUK03EyCKWGQNFgZDA==

```

```
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE
```

```
DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-
```

Sambaattributes.ldf

```
dn: CN=sambaLMPassword,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaLMPassword
attributeID: 1.3.6.1.4.1.7165.2.1.24
attributeSyntax: 2.5.5.4
cn: sambaLMPassword
distinguishedName: CN=sambaLMPassword,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
IDAPDisplayName: sambaLMPassword
name: sambaLMPassword
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE
```

```
dn: CN=sambaNTPassword,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaNTPassword
attributeID: 1.3.6.1.4.1.7165.2.1.25
attributeSyntax: 2.5.5.4
cn: sambaNTPassword
distinguishedName: CN=sambaNTPassword,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
IDAPDisplayName: sambaNTPassword
name: sambaNTPassword
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE
```

```
dn: CN=sambaAcctFlags,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaAcctFlags
attributeID: 1.3.6.1.4.1.7165.2.1.26
attributeSyntax: 2.5.5.4
cn: sambaAcctFlags
distinguishedName: CN=sambaAcctFlags,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
IDAPDisplayName: sambaAcctFlags
name: sambaAcctFlags
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE
```

```
dn: CN=sambaPwdLastSet,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaPwdLastSet
```

```

attributelD: 1.3.6.1.4.1.7165.2.1.27
attributeSyntax: 2.5.5.9
cn: sambaPwdLastSet
distinguishedName: CN=sambaPwdLastSet,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaPwdLastSet
name: sambaPwdLastSet
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaPwdCanChange,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaPwdCanChange
attributelD: 1.3.6.1.4.1.7165.2.1.28
attributeSyntax: 2.5.5.9
cn: sambaPwdCanChange
distinguishedName: CN=sambaPwdCanChange,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaPwdCanChange
name: sambaPwdCanChange
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaPwdMustChange,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaPwdMustChange
attributelD: 1.3.6.1.4.1.7165.2.1.29
attributeSyntax: 2.5.5.9
cn: sambaPwdMustChange
distinguishedName: CN=sambaPwdMustChange,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaPwdMustChange
name: sambaPwdMustChange
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaLogonTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaLogonTime
attributelD: 1.3.6.1.4.1.7165.2.1.30
attributeSyntax: 2.5.5.9
cn: sambaLogonTime
distinguishedName: CN=sambaLogonTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaLogonTime
name: sambaLogonTime
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaLogoffTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaLogoffTime

```

```

attributelD: 1.3.6.1.4.1.7165.2.1.31
attributeSyntax: 2.5.5.9
cn: sambaLogoffTime
distinguishedName: CN=sambaLogoffTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaLogoffTime
name: sambaLogoffTime
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaKickoffTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaKickoffTime
attributelD: 1.3.6.1.4.1.7165.2.1.32
attributeSyntax: 2.5.5.9
cn: sambaKickoffTime
distinguishedName: CN=sambaKickoffTime,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaKickoffTime
name: sambaKickoffTime
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaHomeDrive,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaHomeDrive
attributelD: 1.3.6.1.4.1.7165.2.1.33
attributeSyntax: 2.5.5.4
cn: sambaHomeDrive
distinguishedName: CN=sambaHomeDrive,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaHomeDrive
name: sambaHomeDrive
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE

dn: CN=sambaLogonScript,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaLogonScript
attributelD: 1.3.6.1.4.1.7165.2.1.34
attributeSyntax: 2.5.5.4
cn: sambaLogonScript
distinguishedName: CN=sambaLogonScript,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaLogonScript
name: sambaLogonScript
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE

dn: CN=sambaProfilePath,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaProfilePath

```

```
attributeID: 1.3.6.1.4.1.7165.2.1.35
attributeSyntax: 2.5.5.4
cn: sambaProfilePath
distinguishedName: CN=sambaProfilePath,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaProfilePath
name: sambaProfilePath
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE

dn: CN=sambaUserWorkstations,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaUserWorkstations
attributeID: 1.3.6.1.4.1.7165.2.1.36
attributeSyntax: 2.5.5.4
cn: sambaUserWorkstations
distinguishedName: CN=sambaUserWorkstations,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaUserWorkstations
name: sambaUserWorkstations
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE

dn: CN=sambaHomePath,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaHomePath
attributeID: 1.3.6.1.4.1.7165.2.1.37
attributeSyntax: 2.5.5.4
cn: sambaHomePath
distinguishedName: CN=sambaHomePath,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaHomePath
name: sambaHomePath
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 20
showInAdvancedViewOnly: TRUE

dn: CN=sambaDomainName,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaDomainName
attributeID: 1.3.6.1.4.1.7165.2.1.38
attributeSyntax: 2.5.5.12
cn: sambaDomainName
distinguishedName: CN=sambaDomainName,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaDomainName
name: sambaDomainName
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 64
showInAdvancedViewOnly: TRUE

dn: CN=sambaSID,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaSID
```

```

attributelD: 1.3.6.1.4.1.7165.2.1.20
attributeSyntax: 2.5.5.12
cn: sambaSID
distinguishedName: CN=sambaSID,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaSID
name: sambaSID
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 64
showInAdvancedViewOnly: TRUE

dn: CN=sambaPrimaryGroupSID,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaPrimaryGroupSID
attributelD: 1.3.6.1.4.1.7165.2.1.23
attributeSyntax: 2.5.5.12
cn: sambaPrimaryGroupSID
distinguishedName: CN=sambaPrimaryGroupSID,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaPrimaryGroupSID
name: sambaPrimaryGroupSID
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 64
showInAdvancedViewOnly: TRUE

dn: CN=sambaGroupType,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaGroupType
attributelD: 1.3.6.1.4.1.7165.2.1.19
attributeSyntax: 2.5.5.9
cn: sambaGroupType
distinguishedName: CN=sambaGroupType,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaGroupType
name: sambaGroupType
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaNextUserRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaNextUserRid
attributelD: 1.3.6.1.4.1.7165.2.1.21
attributeSyntax: 2.5.5.9
cn: sambaNextUserRid
distinguishedName: CN=sambaNextUserRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaNextUserRid
name: sambaNextUserRid
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaNextGroupRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaNextGroupRid

```

```

attributelD: 1.3.6.1.4.1.7165.2.1.22
attributeSyntax: 2.5.5.9
cn: sambaNextGroupRid
distinguishedName: CN=sambaNextGroupRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaNextGroupRid
name: sambaNextGroupRid
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaNextRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaNextRid
attributelD: 1.3.6.1.4.1.7165.2.1.39
attributeSyntax: 2.5.5.9
cn: sambaNextRid
distinguishedName: CN=sambaNextRid,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaNextRid
name: sambaNextRid
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

dn: CN=sambaAlgorithmicRidBase,CN=Schema,CN=Configuration,DC=testnet,DC=com
changetype: add
adminDisplayName: sambaAlgorithmicRidBase
attributelD: 1.3.6.1.4.1.7165.2.1.40
attributeSyntax: 2.5.5.9
cn: sambaAlgorithmicRidBase
distinguishedName: CN=sambaAlgorithmicRidBase,CN=Schema,CN=Configuration,DC=testnet,DC=com
instanceType: 4
isSingleValued: TRUE
LDAPDisplayName: sambaAlgorithmicRidBase
name: sambaAlgorithmicRidBase
objectCategory: CN=Attribute-Schema,CN=Schema,CN=Configuration,DC=testnet,DC=com
objectClass: attributeSchema
oMSyntax: 2
showInAdvancedViewOnly: TRUE

DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-
```

SambaClasses.ldf

```

dn: CN=sambaDomain,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaDomain
cn: sambaDomain
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaDomain,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.4.1.7165.2.2.5
instanceType: 4
```

```

IDAPDisplayName: sambaDomain
mayContain: sambaNextRid
mayContain: sambaNextGroupRid
mayContain: sambaNextUserRid
mayContain: sambaAlgorithmicRidBase
mustContain: sambaDomainName
mustContain: sambaSID
name: sambaDomain
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: group
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
rDNAttID: sambaDomainName
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE

dn: CN=sambaldmapEntry,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaldmapEntry
cn: sambaldmapEntry
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaldmapEntry,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.4.1.7165.1.2.2.8
instanceType: 4
IDAPDisplayName: sambaldmapEntry
mayContain: uidNumber
mayContain: gidNumber
mustContain: sambaSID
name: sambaldmapEntry
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: group
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
possSuperiors: sambaDomain
rDNAttID: sambaSID
showInAdvancedViewOnly: TRUE
subClassOf: top
systemOnly: FALSE

dn: CN=sambaSidEntry,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaSidEntry
cn: sambaSidEntry
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaSidEntry,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.4.1.7165.1.2.2.9
instanceType: 4
IDAPDisplayName: sambaSidEntry
mustContain: sambaSID
name: sambaSidEntry
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com

```

```
objectClass: classSchema
objectClassCategory: 1
possSuperiors: group
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
rDNAttID: sambaSID
showInAdvancedViewOnly: TRUE
subClassOf: sambaldmapEntry
systemOnly: FALSE

dn: CN=sambaSamAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaSamAccount
cn: sambaSamAccount
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaSamAccount,CN=Schema,CN=Configuration,dc=testnet,dc=com
governsID: 1.3.6.1.4.1.7165.2.2.6
instanceType: 4
IDAPDisplayName: sambaSamAccount
subClassOf: posixAccount
mayContain: sambaLMPassword
mayContain: sambaNTPassword
mayContain: sambaPwdLastSet
mayContain: sambaLogonTime
mayContain: sambaLogoffTime
mayContain: sambaKickoffTime
mayContain: sambaPwdCanChange
mayContain: sambaPwdMustChange
mayContain: sambaAcctFlags
mayContain: displayName
mayContain: sambaHomePath
mayContain: sambaHomeDrive
mayContain: sambaLogonScript
mayContain: sambaProfilePath
mayContain: description
mayContain: sambaUserWorkstations
mayContain: sambaPrimaryGroupSID
mayContain: sambaDomainName
mustContain: objectClass
mustContain: uid
mustContain: sambaSID
name: sambaSamAccount
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
possSuperiors: sambaDomain
rDNAttID: uid
showInAdvancedViewOnly: TRUE
systemOnly: FALSE

dn: CN=sambaGroupMapping,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaGroupMapping
cn: sambaGroupMapping
```

```

defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaGroupMapping,CN=Schema,CN=Configuration,dc=testnet,dc=com
governorID: 1.3.6.1.4.1.7165.2.2.4
instanceType: 4
IDAPDisplayName: sambaGroupMapping
mayContain: displayName
mayContain: description
mustContain: gidNumber
mustContain: sambaSID
mustContain: sambaGroupType
name: sambaGroupMapping
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: group
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
possSuperiors: sambaDomain
rDNAttID: cn
showInAdvancedViewOnly: TRUE
subClassOf: posixGroup
systemOnly: FALSE

dn: CN=sambaUnixIdPool,CN=Schema,CN=Configuration,dc=testnet,dc=com
changetype: add
adminDisplayName: sambaUnixIdPool
cn: sambaUnixIdPool
defaultSecurityDescriptor:
D:(A;;LCRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;DA)(A;;LCRPRC;;;WD)(A;;CCDCLCS
WRPWPDTLOCRSDRCWDWO;;;SY)
distinguishedName: CN=sambaUnixIdPool,CN=Schema,CN=Configuration,dc=testnet,dc=com
governorID: 1.3.6.1.4.1.7165.1.2.2.7
instanceType: 4
IDAPDisplayName: sambaUnixIdPool
mustContain: uidNumber
mustContain: gidNumber
name: sambaUnixIdPool
objectCategory: CN=Class-Schema,CN=Schema,CN=Configuration,dc=testnet,dc=com
objectClass: classSchema
objectClassCategory: 1
possSuperiors: user
possSuperiors: organizationalPerson
possSuperiors: organizationalRole
possSuperiors: organizationalUnit
possSuperiors: sambaDomain
rDNAttID: ou
showInAdvancedViewOnly: TRUE
subClassOf: organizationalUnit
systemOnly: FALSE

DN:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-

```

Schemaupdate.bat

```
Idifde -i -f Posixattributes.ldf
```

```
ldifde -i -f Posixclasses.ldf  
ldifde -i -f Sambaattributes.ldf  
ldifde -i -f Sambaclasses.ldf
```

Appendix C. Acknowledgements

Portions of the “How Winbind Works” section of this document were directly derived from Chapter 12 “Identity Mapping (IDMAP)” and Chapter 21 “Winbind: Use of Domain Accounts” located in the “The Official Samba-3 HOWTO and Reference Guide” (<http://us1.samba.org/samba/docs/man/Samba-HOWTO-Collection/>).