Debian: SAMBA as PDC with LDAP backend HOWTO

Reto Bürki

August 7, 2006

Abstract

This HOWTO explains the configuration of a Debian (Sarge) based Linux System to act as a Samba PDC with LDAP backend. This same LDAP backend can also be used by Linux clients for authentication information over PAM/NSS. Since the LDAP stored user objects provide all necessary attributes, it is possible to have a central data storage for administration. This makes such a system a perfect solution in heterogeneous networks.

Contents

1 Introduction

Although there are many existing HOWTOs covering the installation of Samba with LDAP backend, I could not find a suitable one which explained everything I needed to know in kind of a short manner. Practically speaking, I was looking for a HOWTO which tells me exactly what to type on the shell to get a Samba PDC up and running quickly on Debian Sarge.

So this HOWTO aims to fill this gap between the blown-up HOWTOs covering a lot of possible configurations on different Linux distributions and the rather short ones with no information at all. Even though setting up a Samba with LDAP is a quite complex task, you should be able to fulfil it with help of this HOWTO by just following all the steps explained in the following chapters. This HOWTO does not claim to be a complete guide for Samba and LDAP, there are other documents with exactly that purpose.

1.1 Copyright Information

1.2 Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility for that.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.3 Credits

Since this is the initial release, not many people have found this HOWTO (yet). I hope to thank someone for his contribution in the future .) we'll see about that.

1.4 Feedback

Feedback is most certainly welcome for this document. Without your submissions and input, this document wouldn't exist. Please send your additions, comments and criticisms to the following email address : buerki (at) swiss-it.ch.

2 Installation

In this HOWTO we'll go through the installation and configuration of every element step by step. Since the LDAP backend is the most important service to run correctly and all other services use it as a base, we'll start with this element. This HOWTO also covers the installation of phpldapadmin, a web based front-end for LDAP directory management. Note that all the explanations here are directly targeting Debian Sarge Linux, although they might apply to other distros as well. Now let's get started. I assume you already have a running Debian in a minimal installation.

2.1 OpenLDAP

In this section we'll install and configure the OpenLDAP daemon. In order to get OpenLDAP working, we need to install some additional packages first. With Debian's `apt` system, this task is rather easy:

```
# apt-get install slapd ldap-utils db4.2-util
```

This will install all the packages and dependencies needed for running the OpenLDAP LDAP daemon (slapd).

Now you'll be asked some questions about how to configure OpenLDAP. you can gladly accept most of Debian's default settings. Just make sure you choose the appropriate base dn for your setup and of course a secure password. I'll use the dn *"dc=swiss-it,dc=ch"* in this HOWTO. After the LDAP daemon is started, the backend for Samba is in place. For Samba to work correctly with `slapd` we need to modify the generated `/etc/ldap/slapd.conf` configuration file.

First, we need to add the Samba schema. In order to do this, we have to install the `samba-doc` package:

```
# apt-get install samba-doc
```

After the samba-doc package is installed, we need to copy the schema to the correct location:

```
# cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz \
    /etc/ldap/schema/
# gunzip /etc/ldap/schema/samba.schema.gz
```

Then add the corresponding include line in `/etc/ldap/slapd.conf`:

```
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/samba.schema
```

Samba needs some special indexes to work properly, so we need to add them as well:

```
index     objectClass,uidNumber,gidNumber   eq
index     cn,sn,uid,displayName             pres,sub,eq
index     memberUid,mail,givenname          eq,subinitial
index     sambaSID,sambaPrimaryGroupSID,sambaDomainName eq
```

To protect the samba attributes used for authentication, add sambaLMPassword and sambaNTPassword to the *"access to"* line in `/etc/ldap/slapd.conf`:

```
access to attrs=userPassword,sambaLMPassword,sambaNTPassword
    by dn="cn=admin,dc=swiss-it,dc=ch" write
    by anonymous auth
    by self write
    by * none
```

That's it for now with OpenLDAP. Restart slapd:

```
# /etc/init.d/slapd restart
```

If you see something like:

```
Stopping OpenLDAP: slapd.
Starting OpenLDAP: running BDB recovery, slapd.
```

you should be on the safe side :) If not, blame this HOWTO (and please send feedback). Except you'd like to work with the ldap command line tools (which are indeed not very comfortable), I would suggest to use the web based front-end for LDAP management, phpldapadmin. The next subsection covers the installation and configuration of phpldapadmin.

2.1.1 phpldapadmin

For phpldapadmin to work, we need to install a php-enabled Apache Webserver:

```
# apt-get install libapache2-mod-php4 php4-dev \
    php4-pear php4-ldap
```

This command installs and starts `apache2` with all dependencies. No further configuration is needed for Apache at this point. Now it's time to install phpldapadmin:

```
# apt-get install phpldapadmin
```

Tell Debian to manage all server configs. Then restart Apache. The important phpldapadmin configuration-files are:

- /usr/share/phpldapadmin/config.php
- /usr/share/phpldapadmin/templates/template_config.php
- (/etc/apache2/conf.d/phpldapadmin)

The last config file is only important if you are not happy with Debian's default Apache2 phpldapadmin configuration (maybe you'd like to enable SSL for example). Modify the other two configs to your needs. The configs are well commented and should be self-explanatory. I'll therefore only point out the most important aspects here.

/usr/share/phpldapadmin/config.php
The following are the most important settings to make phpldapadmin work in our environment. Be sure to set the correct "base" and "login_dn". I'm using "crypt" as default hash algorithm troughout the whole setup. Make sure not to mix the algorithms because this can cause strange errors which are very hard to trace.

```
$servers[$i]['name'] = 'secunet LDAP server';
$servers[$i]['host'] = 'localhost';
$servers[$i]['base'] = 'dc=swiss-it,dc=ch';
$servers[$i]['auth_type'] = 'session';
$servers[$i]['login_dn'] = 'cn=admin,dc=swiss-it,dc=ch';
$servers[$i]['default_hash'] = 'crypt';
```

Listing 1: /usr/share/phpldapadmin/config.php

The rest can be left as it is. Most of the settings are set correctly by Debian's debconf anyway. I don't care about TLS either, because our LDAP server is on the same host as phpldapadmin.

/usr/share/phpldapadmin/templates/template_config.php
This template configuration file affects the way how phpldapadmin creates new entries. It is especially important to configure Samba related settings ("SAMBA TEMPLATE CONFIGURATION"), most important the SID of the machine:

```
array( 'name'  => 'SWISSIT',
       'sid' => 'S-1-5-21-498580034-1323423440-212378881' );
```

You can get the SID by entering (do this after you installed samba, see 2.2):

```
# net getlocalsid
```

or

```
# net rpc info
```

Anything else should be O.K. Now it's time to test OpenLDAP and phpldapadmin. Use your browser and open phpldapadmin. The URL is something like *"http://your-server/phpldapadmin"*. You should be able to login with your LDAP admin (*"cn=admin,dc=swiss-it,dc=ch"* in our example) and password. If not, your log file analysis and debugging session might be ready to begin :) If it works as aspected, avoid making to much mess in the directory (yet) because this was just a test to see if LDAP and phpldapadmin work smoothly together. Log out now and proceed with this HOWTO.

2.2 Samba

Now, let's move on with the installation and configuration of Samba. To get the Software installed, type:

```
# apt-get install samba smbldap-tools cupsys cupsys-bsd
```

This will install Samba on your system. Ignore all questions from Debian's debconf, because we'd like to do all configurations by modifying the necessary configuration files directly. First we start by creating needed directories for Windows Profiles (although Roaming Profiles are a pain in the ass and should be avoided) and Netlogon.

```
mkdir /samba
mkdir /samba/netlogon
mkdir /samba/profiles
chmod 1777 /samba/profiles
mkdir /var/spool/samba
```

The most complex part is to write an appropriate smb.conf file to work with LDAP...and that's why I've done this for you already. How kind of me. Just change the appropriate values to your needs (Workgroup etc.):

```
# Global parameters
[global]
        workgroup = SWISSIT
        netbios name = PDC
        enable privileges = yes
        #interfaces = 192.168.5.11
        username map = /etc/samba/smbusers
        server string = Samba PDC-Server %v
        security = user
        encrypt passwords = Yes
        obey pam restrictions = No
        ldap passwd sync = Yes
        #log level = 3
        syslog = 0
        log file = /var/log/samba/log.%m
        max log size = 50
        time server = No
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
        mangling method = hash2
        Dos charset = 850
        Unix charset = ISO8859-1
        hide dot files = Yes

        logon script = logon.cmd
        logon drive = H:
        logon home =
        logon path =

        domain logons = Yes
        os level = 65
        preferred master = Yes
        domain master = Yes
        wins support = Yes
        passdb backend = ldapsam:ldap://127.0.0.1/
        ldap admin dn = cn=admin,dc=swiss-it,dc=ch
        ldap suffix = dc=swiss-it,dc=ch
        ldap group suffix = ou=groups
        ldap user suffix = ou=users
        ldap machine suffix = ou=workstations
        ldap idmap suffix = ou=users
        #ldap ssl = start tls
        add user script = /usr/sbin/smbldap-useradd -m "%u"
        ldap delete dn = Yes
        #delete user script = /usr/sbin/smbldap-userdel "%u"
```

```
        add machine script = /usr/sbin/smbldap-useradd -g 515 -w "%u"
        add group script = /usr/sbin/smbldap-groupadd -p "%g"
        #delete group script = /usr/sbin/smbldap-groupdel "%g"
        add user to group script = /usr/sbin/smbldap-groupmod -m "%u"
             "%g"
        delete user from group script = /usr/sbin/smbldap-groupmod -x
             "%u" "%g"
        set primary group script = /usr/sbin/smbldap-usermod -g "%g"
            "%u"

        # printers configuration
        printer admin = @"Print Operators"
        load printers = Yes
        create mask = 0640
        directory mask = 0750
        nt acl support = No
        printing = cups
        printcap name = cups
        deadtime = 10
        guest account = nobody
        map to guest = Bad User
        dont descend = /proc,/dev,/etc,/lib,/lost+found,/initrd
        show add printer wizard = yes
        preserve case = yes
        short preserve case = yes
        case sensitive = no

[homes]
        comment = Home Directories
        read only = No
        create mask = 0644
        directory mask = 0775
        browseable = No

[netlogon]
        path = /samba/netlogon/
        browseable = No
        read only = yes

[profiles]
        path = /samba/profiles
        read only = no
        create mask = 0600
        directory mask = 0700
        browseable = No
        guest ok = Yes
        profile acls = yes
        csc policy = disable
        # next line is a great way to secure the profiles
```

```
        force user = %U
        # next line allows administrator to access all profiles
        valid users = %U @"Domain Admins"

[printers]
        comment = Network Printers
        printer admin = @"Print Operators"
        guest ok = No
        printable = yes
        path = /tmp
        browseable = No
        read only = Yes
        #print command = /usr/bin/lpr -P%p -r %s
        #lpq command = /usr/bin/lpq -P%p
        #lprm command = /usr/bin/lprm -P%p %j

[print$]
        path = /var/lib/samba/printers
        guest ok = No
        browseable = Yes
        read only = No
        valid users = @"Print Operators",@"Domain Admins"
        write list = @"Print Operators",@"Domain Admins"
        create mask = 0664
        directory mask = 0775
```

Listing 2: /etc/samba/smb.conf

Next, create the file "/etc/samba/smbusers" with the following content:

```
root = Administrator
```

Now, we need to configure the `smbldap-tools`, the tools which interact with LDAP on behalf of Samba. Two configuration files are important:

- /etc/smbldap-tools/smbldap_bind.conf
- /etc/smbldap-tools/smbldap.conf

/etc/smbldap-tools/smbldap_bind.conf
This configuration file defines how the tools bind to the LDAP-Server. In my configuration, this file has the following entries:

```
masterDN="cn=admin,dc=swiss-it,dc=ch"
masterPw="you_wish"
```

Since this file includes our LDAP admin password in plain-text, make sure to make it readable only by root:

```
# chmod 600 /etc/smbldap-tools/smbldap_bind.conf
```

/etc/smbldap-tools/smbldap.conf
This configuration file is very important for our setup to work correctly. Any mistakes (and typos) made here can produce sleepless nights crawling through log-files :) The most important settings are:

```
# 'net getlocalsid' again
SID="S-1-5-21-498580034-1323423440-212378881"

slaveLDAP="127.0.0.1"
slavePort="389"
masterLDAP="127.0.0.1"
masterPort="389"

suffix="dc=swiss-it,dc=ch"
usersdn="ou=users,${suffix}"
computersdn="ou=workstations,${suffix}"
groupsdn="ou=groups,${suffix}"
idmapdn="ou=idmap,${suffix}"
sambaUnixIdPooldn="sambaDomainName=SWISSIT,${suffix}"
scope="sub"
# !!!
hash_encrypt="CRYPT"
crypt_salt_format="%s"
ldapTLS=0

userLoginShell="/bin/bash"
userHome="/home/%U"
userGecos="System User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"


################################################
#
# SAMBA Configuration
#
################################################
# replace PDC with your server netbios name
userSmbHome="\\PDC\%U"
userProfile="\\PDC\profiles\%U"
userHomeDrive="H:"
```

```
userScript="logon.cmd"
mailDomain="swiss-it.ch"
```

Listing 3: /etc/smbldap-tools/smbldap.conf

Samba needs to know the password to contact the LDAP-Server. You can provide this information by typing

```
# smbpasswd -w you_whish
```

on the command line. Now, if you are very brave, it's time to "populate" all the information needed by Samba to operate with LDAP. To do this, simply type:

```
# smbldap-populate
```

This should create all entries in the LDAP-directory. If not, it's time for a little error searching, time consuming debugging-session...or you could write an angry mail about this crappy HOWTO to the author :)

If you are still reading, everything should have worked as planned. If you like, you can verify and enjoy your work by browsing the LDAP-directory with phpldapadmin. The next step is now to inform Linux about our new shiny LDAP-directory.

2.3 PAM and friends

The user objects we'll create in the LDAP-directory can also be used by Linux for authentication since such objects consist of *posixAccount*, *shadowAccount* and *sambaSamAccount* object classes. This is very comfortable, since our PDC can act as authentication source for Windows and Linux clients, what makes it very ideal in heterogeneous environments. Before we go any deeper into the Samba configuration, let's test all the settings affecting Linux. These settings are also a pre-requisite for Samba. First, we need to install all modules and software:

```
# apt-get install libpam-ldap libnss-ldap libpam-cracklib
```

Debian's debconf will ask some questions again, you can gladly choose the default values since we'll modify the generated configuration files anyway. The first step is to inform the *name service switch* (NSS) about the new information source. Modify `/etc/nsswitch.conf`:

```
passwd:         files ldap
```

```
group:          files ldap
shadow:         files ldap
```

Listing 4: /etc/nsswitch.conf

Now, we need to inform the name switching service how to obtain the needed information from the directory. This is done by modifying the appropriate config-file, `/etc/libnss-ldap.conf` in this case. Mine looks like this:

```
host 127.0.0.1
base dc=swiss-it,dc=ch
ldap_version 3
scope one
pam_filter objectclass=posixaccount
pam_login_attribute uid
pam_member_attribute gid
pam_password crypt

nss_base_passwd ou=users,dc=swiss-it,dc=ch?one
nss_base_passwd ou=workstations,dc=swiss-it,dc=ch?one
nss_base_shadow ou=users,dc=swiss-it,dc=ch?one
nss_base_group ou=groups,dc=swiss-it,dc=ch?one
```

Listing 5: /etc/libnss-ldap.conf

Copy this file to `/etc/pam_ldap.conf` as well or create a symlink:

```
# rm -rf /etc/pam_ldap.conf
# ln -s /etc/libnss-ldap.conf /etc/pam_ldap.conf
```

Some changes are required to the PAM module configurations, because the default Debian configs are not very suitable. Change the files in `/etc/pam.d` according to the following:

```
auth    required    pam_env.so
auth    required    pam_tally.so deny=20
auth    sufficient pam_unix.so likeauth nullok shadow
auth    sufficient pam_ldap.so use_first_pass
auth    required    pam_deny.so
```

Listing 6: /etc/pam.d/common-auth

The `pam_tally.so` module locks the user on a client after 10 unsuccessful login attempts. To unlock the account on this client, root has to type this command:

```
# pam_tally --user the_user --reset
```

This will enable the account by resetting the false login count to 0. Somehow `pam_tally.so` counts one unsuccessful login as two, so deny=20 actually means 10 unsuccessful login attempts. If someone could point out what I'm doing wrong here, thanks :)

```
account requisite pam_unix.so
account sufficient pam_localuser.so
account required pam_ldap.so
# account required pam_tally.so
```

Listing 7: /etc/pam.d/common-account

NOTE: `pam_localuser.so` is not yet included in Debian Sarge (stable). I had to get it from testing/unstable. This module is used to check whether a user exists locally (in `/etc/passwd`). The pam_tally.so module is commented out because it did not work as expected. No non-root user could log in and there were strange errors ("Tally underflowed for user x") in `/var/log/auth.log`. This has of course consequences: Even though *pam_tally.so* ensures to deny login when the user exceeded the maximum login attempts, the account is still valid and could be abused (e.g. authentication data for services like cyrus and so on). I've tried this whole tally thing on a clean base install of Debian/Testing by just adding the two lines into common-auth and common-account. Same effect. Solutions are appreciated.

```
password   required  pam_cracklib.so retry=3 minlen=12 difok=4
password   sufficient pam_unix.so remember=4 nullok use_authtok
    shadow md5
password   sufficient pam_ldap.so use_authtok use_first_pass
password   required  pam_deny.so
```

Listing 8: /etc/pam.d/common-password

To enable the password history code of pam_unix.so, we also need to create an empty file for password hash storage:

```
# touch /etc/security/opasswd
# chown root:root /etc/security/opasswd
# chmod 600 /etc/security/opasswd
```

This would be it in theory for the password reminder function. But since we use LDAP for user storage, it does not work :) → I'm looking for a solution here...

At this point you may be wondering how to get the system to automatically force users to change their password after some period of time. These parameters are normally set in the `/etc/login.defs` file. But since we use

LDAP here, pam reads the required attributes directly from the user object stored on the LDAP server. The important attributes are:

- *shadowLastChange*
  date/time of the user's last password change
- *shadowMax*
  how often users have to change their passwords
- *shadowMin*
  how long a user is forced to live with their new password before their allowed to change it again
- *shadowWarning*
  number of days before the password expiration date that the user is warned that their password is about to expire

To force the users to change their passwords regularly, let's use these values:

```
shadowMax      90
shadowMin       7
shadowWarning   7
```

The user is required to change his password after 90 days. The password must be 7 days old before it can be changed again. Display a warning message about password expiration 7 days before. To administratively force the user to change his password at the next login, set *shadowLastChange* to 0.

```
session required  pam_limits.so
session required  pam_unix.so
session required  pam_mkhomedir.so skel=/etc/skel/ umask=0066
session optional  pam_ldap.so
```

Listing 9: /etc/pam.d/common-session

Client Configuration

I won't cover client installation in detail here because it's actually the same as on the server. All you need to change is the *host* entry in your `/etc/libnss-ldap.conf` (and `pam_ldap.conf` accordingly). There is one thing though: If you want to enforce the described password policy under linux, you have to add the following entry in your `/etc/pam_ldap.conf`:

```
rootbinddn cn=admin,dc=swiss-it,dc=ch
```

Then, write your LDAP password to `/etc/ldap.secret`:

```
# echo "you_wish" > /etc/ldap.secret
# chmod 600 /etc/pam_ldap.secret
# ln -s /etc/pam_ldap.secret /etc/ldap.secret
```

pam_ldap needs this entry to change the *shadowLastChange* attribute when the user changes his password. Giving write permission to this attribute is of course not a good idea if you'd like to ENFORCE a strong password policy :)

TLS

TODO.

3 Testing and Configuration

Time for a first test of the environment...and of course, debugging. I don't think this setup will work as is and this HOWTO of course needs a lot of improvements. So don't hesitate to send feedback, corrections to *buerki (at) swiss-it.ch*. Before we begin, restart every service first:

```
# /etc/init.d/slapd restart
# /etc/init.d/samba restart
```

3.1 Linux and PAM

Issue the following command:

```
# smbldap-useradd -m -a test
# smbldap-passwd test
```

This should create a user *test* in "ou=users,dc=swiss-it,dc=ch" (in my example) with password "test". This user is a Windows user, which means it can login on Linux and Windows clients. If this command returns no error, verify the existence of this user in the Linux environment:

```
# getent passwd
```

You should now see the user "test" listed:

```
test:x:1001:513:System User:/home/test:/bin/bash
```

If not, Linux does not know about the existence of this user and you cannot proceed with the next step. In this case, verify the existence of the user object in the LDAP-directory first (with phpldapadmin for example) and narrow down the problem. If the user exists in the directory, something

with PAM and NSS does not work as expected. Check `/var/log/auth.log` for details.

If the user is listed how it should be, you can now login with this user on a console. This should just work. If not, examine `/var/log/auth.log` and `/var/log/syslog`. If you don't see anything, increase the "loglevel" of `slapd` by putting "loglevel 256" in `/etc/ldap/slapd.conf`. "256" is a good value for debugging as I found out :), enter `man slapd.conf` for more values of this parameter.

Make sure LDAP user authentification works on Linux before proceeding to play with Samba.
NOTE: You have to restart the SSH daemon in order to make the new settings work with SSH.

3.2 Samba

Domain Administrator

Samba needs a "Domain Administrator" with the power to control the domain. Create this user with the following command:

```
# smbldap-useradd -c "SMB root" -a -m -g 0 -u 0 root
# smbldap-passwd root
```

This user has all rights in the given Domain and should only be used for initial setup. If you login onto a Windows workstation with the username "Administrator" (SWISSIT
Administrator'), Samba will automaticall map this user to the "root" user we just created. You should disable this user after initial setup with the following command:

```
# smbpasswd -d root
```

To re-enable the almighty root (Administrator) user in the domain, just issue this command:

```
# smbpasswd -e root
```

Domain Admins

It is much better and safer to add a "normal" user to the domain's "Domain Admins" group. To grant the members of the Domain Admins group on a Samba domain controller, the capability to add client machines to the domain, you need to type the following:

```
# net rpc rights grant 'SWISSIT\Domain Admins' \
    SeMachineAccountPrivilege
```

Then add your privileged user to the "Domain Admins" group, for example with phpldapadmin or with the smbldap-tools.

Group Mappings

Samba uses special group mappings to associate Unix (POSIX) groups with Windows groups. It is therefore possible to "map" the "Domain Admins" group for example to a specific Unix group, lets say "smbadmins". All members of this Unix group are then automatically members of the Windows group within that domain. It is very easy to declare Samba group mappings in phpldapadmin. You can list the actual mappings with:

```
# net groupmap list
```

Print Operators

We have a special "Print Operators" group for printer management. These group must have the appropriate privilege to do this job:

```
# net rpc rights grant 'SWISSIT\Print Operators' \
    SePrintOperatorPrivilege
```

You should see something like "successfully granted rights". If you see something else, throw the server out of the window.

If everything worked fine, we verified that all the settings are correct and your domain is up and running.

3.3 Windows Clients

This is where the real fun begins. On Windows XP (and Windows 2000?) Clients, fire up the "Group Policy Editor" (`gpedit.msc`) and enable

```
Computer Configuration
 \____>Administrative Templates
        \____>System
               \____>User Profiles
                      * Do not check for user ownership
                        of Roaming Profile Folders
```

That's it in theory! In practice, other nice traps can come into play. I would appreciate any feedback on "Windows Clients" and this little HOWTO.

You can add your workstation inside "System Preferences" → "System". Only the domain administrator (root) or members of the group "Domain Admins" are allowed to add and remove workstations to the domain.

3.4 Printing

...or is this real fun? Has to be written.

3.5 Password Policy

As an example, we'll enable a strict password policy on our domain:

1. min password length = 8 characters
2. password history = save last 4 passwords
3. maximum password age = 90 days
4. minimum password age = 7 days
5. lockout after 10 attempts
6. lockout duration = forever, account must be manually reenabled

To enable this policy, execute the following commands on your PDC:

```
# pdbedit -P "min password length" -C 8
# pdbedit -P "password history" -C 4
# pdbedit -P "maximum password age" -C 7776000
# pdbedit -P "minimum password age" -C 604800
# pdbedit -P "bad lockout attempt" -C 10
# pdbedit -P "lockout duration" -C -1
```

4 Links

You'll find anything you might possibly need at this location:
http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/

5 FAQ

Q should I use `nscd`, the name service caching deamon?

A I would not recommend this during roll out although it would increase the pleasure you have finding errors and misconfigurations. It makes sense in a productive environment though to speed up user queries.

Q What kind of hash should I use with POSIX attribute "userPassword"?

A You should use CRYPT as hash. Had problems with other hashes and linux clients. at least make sure all your settings are consistent and use the same hash algorithm.