The SAMBA-2.2.4/LDAP PDC HOWTO

Olivier Lemaire

Revision : 1.24, generated June 7, 2002

Contents

[1] http://IDEALX.com/

1 Introduction

I hope this document can help: it express our personal experience using Samba[2] and OpenLDAP[3] together to replace Microsoft Windows NT PDC (Primary Domain Controler).

This howto currently runs for :

- Samba-2.2.4,
- Microsoft Windows, Microsoft Windows NT 4 and Microsoft Windows 2000 Workstations,
- *Linux* RedHat 7.2 (should work on any *Linux* distro anyway [4]),
- OpenLDAP 2.0.x (at least 2.0.11, we used 2.0.21 in this howto)

The last release (most up to date) of this document may be found on the http://samba.idealx.org/ project page.

[2] http://www.samba.org

[3] http://www.OpenLDAP.org/

[4] some special Debian notes are provided for Woody

2 Context of this Howto

This Howto aims at helping to configure an `Samba` + `OpenLDAP` Primary Domain Controler for Microsoft Windows Workstations (and, using nss_ldap and pam_ldap, a unique source of authentification for all workstations, including *Linux* and other Unix systems).

For the need of our example, we settled the following context :

- All workstations and servers are in the same LAN 192.168.1.0/24,
- DNS resolution is okay (using `Bind` or `Djbdns` for example), and out of the scope of this Howto [5],
- We want to configure the Microsoft Windows NT Domain named **IDEALX-NT**,
- We will have a central Primary Domain Controler named **PDC-SRV** (netbios name) on the host 192.168.1.1/32 ,
- We want this Primary Domain Controller to be the WINS server and the Master Browser Server of the IDEALX-NT domain,
- All authentifications objects (users and groups) will be stored on an `OpenLDAP` server, using the base DN : **dc=IDEALX,dc=ORG**,
- `Samba`[6] Users accounts will be stored in **ou=Users,dc=IDEALX,dc=ORG**,
- `Samba` Computers accounts will be stored in **ou=Computers,dc=IDEALX,dc=ORG**,
- `Samba`[7] Groups accounts will be stored in **ou=Groups,dc=IDEALX,dc=ORG**,

Separating `Samba` accounts (Users and Computers) and Groups is a optional way to do the job. We could store all this datas under the same DN, but we made this distinction to make the LDAP tree more human-readable[8]. Feel free to change those statements (Microsoft Windows NT Domain Name, LDAP tree) for a context who feet you better, if desired.

In this Howto, we took the RedHat *Linux* 7.2 as a base, and tried to conform to FHS [9] recommandations. All RPMS and SRPMS packages for RedHat *Linux* 7.2 are available on the `http://samba.idealx.org/` project page. This do not mean `Samba` only work on RedHat *Linux* of course (nor only on *Linux* for short), but just that this choice present the advantage to be quickly reproductible by anybody (RedHat *Linux* is very common on the server market nowadays, and supported by many vendors).

[5] DNS resolution **must** be ok to use Samba without spending hours trying to understand why that think is supposed to work and don't !

[6] and other Posix accounts so the PDC will provide an unique source of authentification for Windows and Unix stations

[7] and other Posix groups so the PDC will provide an unique source of system datas for Windows and Unix stations

[8] additionnaly, there is a potential issue with computer management via LDAP : see 10 on page 28

[9] see `http://www.pathname.com/fhs/` for more info on FHS

We took care about FHS recommandations as we want to be able to make this PDC Higly Available (in a futur revision of this Howto), and wish to seperate system and addons software (and for that, FHS is a good text to follow).

3 Download & compile

To stick to this Howto[10], you must have the following requirements prior to download anything :

- RedHat *Linux* 7.2 installed and operational (network included),
- you must be prepared (if not already done) to use pam_ldap and nss_ldap (we'll see later how to configure them correctly).

Additionnaly, you must download :

- `Samba` release 2.2.4 (see below),
- `OpenLDAP` release 2.0.11 or 2.0.21 (see below),
- `nss_ldap` and `pam_ldap` (see below),
- `smbldap-tools` release 0.7 (see below).

3.1 `OpenLDAP` 2.0.21

At the date we wrote this document, release 2.0.21 of `OpenLDAP` was considered stable enought to be used in production environment. We tested it (see 16.1 on page 41), and everything was ok, so we used it.

Just download some of the following packages :

- openldap-2.0.21-1, openldap-servers-2.0.21-1, openldap-clients-2.0.21-1 and nss_ldap-172-2 packages from RedHat *Linux* 7.2, if you want to stick to RedHat 7.2 packages,
- or openldap-2.0.23-1, openldap-servers-2.0.23-1, openldap-clients-2.0.23-1 and nss_ldap-173-3 packages from RawHide RedHat, if you want to run a 2.0.23 release of `OpenLDAP`.

You're free to use release 2.0.21 of `OpenLDAP` : we tested it and everything was ok with `Samba`. However, lots of bugfixes were added to `OpenLDAP` in the 2.0.23 release, so you're encouraged to use this release instead.

For production purpose, we used `OpenLDAP` release 2.0.23 and we encourage you to use the same release.

RPMs for `OpenLDAP` release 2.0.23 may be found at `ftp://ftp.redhat.com/pub/redhat/linux/rawhide/i386/RedHat/RPMS/`, and you will need to install the following packages:

```
openldap-2.0.23-1.i386.rpm
openldap-clients-2.0.23-1.i386.rpm
openldap-servers-2.0.23-1.i386.rpm
```

[10] remember: feel free to test under other distros and OS, and please report : we'll update this Howto

3.2 Samba 2.2.4

Samba 2.2.4 is the last release of Samba2.2 branch (at the date of this Howto redaction, and used by this Howto). To use it with LDAP, some patches must be added to the base release.

In this Howto, we used the RedHat RawHide package as a base (patches are already included, thank's to RedHat RawHide team), and rebuild the package for LDAP support :

- grab the samba-2.2.4-1.src.rpm from RawHide (ftp://ftp.redhat.com/pub/redhat/linux/rawhide/SRPMS/SRPMS/samba-2.2.4-1.src.rpm),
- install it on you system (rpm -ivh samba-2.2.4-1.src.rpm),
- edit your /usr/src/redhat/SPECS/samba.spec to add the following configure options : –with-acl-support –with-profile –disable-static –with-msdfs –with-ldapsam, by the way, edit the Release tag to update it to '2' (Release: 2),
- build the RPMS (cd /usr/src/redhat/SPECS/ && rpm -ba samba.spec),
- then install the following RPMS : samba-common-2.2.4-2, samba-client-2.2.4-2 and samba-2.2.4-2)

You'll find those Samba packages already prepared in the SAMBA-LDAP projet page (see http://samba.idealx.org/).

You will need to install the following packages:

```
samba-common-2.2.4-2
samba-client-2.2.4-2
samba-2.2.4-2
```

3.3 smbldap-tools

smbldap-tools is a package containing some useful scripts to manage users/groups when you're using LDAP as source of users/groups datas (for Unix and for Samba). We used those scripts in this Howto to add/del/mod users and groups.

Those scripts are under packaging at June 7, 2002.

For now, just grab the smbldap-tools-0.7.tgz and detar/save the tools under : /usr/local/sbin/.

Alternatively, you can use the smbldap-tools RedHat package provided at http://samba.idealx.org/dist/redhat/.

4 Configuring OpenLDAP

You'll need to configure your OpenLDAP server to serve as SAM database for Samba-2.2.4. Following our context example, we must to configure it to :

- accept the Samba-2.2.4 LDAP v3 schema,
- run on the base DN dc=IDEALX,dc=ORG,
- contain the minimal entries needed to start using it.

For the needs of this HOWTO example, we have used the following LDAP tree :

```
(using Relative DN notation)

dc=IDEALX,dc=ORG
 |
 `--- ou=Users  :    to store user accounts (both posixAccount and
 |                   sambaAccount) for Unix and Windows systems
 |
 `--- ou=Computers : to store computer accounts (sambaAccount) for Windows
 |                   systems
 |
 `--- ou=Groups :   to store system groups (posixGroup) for Unix and Windows
                    systems (or for any other LDAP-aware systems)
```

You may choose to use another LDAP tree to store objects : for example, all accounts (shadowAccounts and sambaAccounts) "under" the same DN. We tought it was simplier to understand like this (and was not a problem for an Unix-nss_ldap do deal with).

Additionnaly, using shadowAccount is not mandatory : if you don't use shadow password on you Unix systems, you should use posixAccounts instead.

Using Samba-2.2.4 and OpenLDAP, we will store :

- Windows user accounts using sambaAccount object class (samba.schema),
- Windows computer accounts (ie. workstations) using sambaAccount object class,
- Unix-only user accounts using shadowAccount object class (nis.schema) [11],
- Users groups (Windows and Unix, as it seems there is no difference in Samba release 2.2.4[12] using posixGroup object class.

[11] as we already saw, using shadowAccount is not mandatory : if you don't use shadow suite passwords, you just need posixAccount

[12] It's not the same using SAMBA-TNG, who use sambaGroups and other specific object classes

4.1 Schemas

First, copy the Samba samba.schema to /etc/openldap/schema/samba.schema.

You'll find this Samba schema shipped with the Samba-2.2.4 release (/example/LDAP/samba.schema in the source package, or in /usr/share/doc/samba-2.2.4/examples/LDAP/samba.schema if you used the modified RedHat RawHide package to build and install Samba)

If you plan using inetOrgPerson schema, then edit this schema to comment the 'displayName' attributetype. In this Howto, we'll use inetOrgPerson schema who already define this attributetype. You can have a look on 22.1 on page 51 to see a sample 'patched' Samba schema. If you don't use inetOrgPerson, then you don't need to comment the 'displayName' in the samba.schema. In this Howto we've used inetOrgPerson because we want to merge organizational datas with technical datas, in a technical directory. It's not mandatory : feel free to use a context who feet your needs.

4.2 Configuration

Create your /etc/openldap/slapd.conf to configure your server :

```
# /etc/openldap/slapd.conf file for SAMBA-LDAP

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

database        ldbm
suffix          "dc=IDEALX,dc=ORG"
rootdn          "cn=Manager,dc=IDEALX,dc=ORG"
rootpw          secret
directory       /var/lib/ldap

index   objectClass,rid,uid,uidNumber,gidNumber,memberUid   eq
index   cn,mail,surname,givenname                                 eq,subinitial

# - The End
```

Then, edit your /etc/openldap/ldap.conf to indicate your base DN and default server:

```
# /etc/openldap/ldap.conf for samba-ldap
#
# LDAP Defaults

HOST 127.0.0.1
BASE dc=IDEALX,dc=ORG

# - The End
```

Finally, start your OpenLDAP server : /etc/init.d/ldap start. Everything should work fine. If not :

- verify your schemas,

- verify that /var/lib/ldap exist and is owned by the user who run sladp (ldap user for RedHat OpenLDAP packages),
- consult the OpenLDAP documentation.

4.3 Initial entries

Next, we'll inject some initial entries on the brand new OpenLDAP server configured and started above.

A sample LDIF file is presented on 22.2 on page 53. copy/paste it on a file named base.ldif and add it using:

```
ldapadd -x -h localhost -D "cn=manager,dc=IDEALX,dc=ORG" -f base.ldif -W
```

(type your admin DN password, 'secret' to complete the command)

4.4 smbldap-tools configuration

Finally, you must configure your smblda-tools to match your system and LDAP configuration : edit the /usr/local/sbin/smbldap_conf.pm and configure it according to your LDAP configuration (RootDN password and LDAP server @IP address).

You'll find two confusing entry: slaveLDAP and masterLDAP. For our first example, those two LDAP server will be the same one, but in a real life configuration, you may want to have a slave server to serve all your read request, and one dedicated to write request. Anyway, in the current example, as we build the PDC using Samba and OpenLDAP on the same host, you should specify 127.0.0.01 for the two LDAP servers.

You'll find some other configuration options in this configuration file: those are the default values used by smbldap-tools when creating an account (user or computer). Feel free to change those values if desired.

5 Configuring Linux

You need to tell you *Linux* box to use LDAP (pam_ldap and nss_ldap). Then, you should run `nscd` and finish your system LDAP configuration.

5.1 pam_ldap, nss_ldap and nscd

Use 'authconfig'[13] to activate pam_ldap :

- Cache Information
- Use LDAP
- dont select 'Use TSL'
- Server: 127.0.0.1
- Base DN: dc=IDEALX,dc=ORG
- Use Shadow Passwords
- Use MD5 Passwords
- Use LDAP Authentification
- Server : 127.0.0.1
- Base DN: dc=IDEALX,dc=ORG

Cache Information mean you're using nscd (man nscd for more info) : if you're going to use pam_ldap and nss_ldap, you should really use it for optimization.

If you don't rely on 'authconfig', you can edit your /ets/pam.d/system-auth by hands, to have something like the following:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/pam_ldap.so use_first_pass
auth        required      /lib/security/pam_deny.so

account     required      /lib/security/pam_unix.so
account     sufficient    /lib/security/pam_ldap.so

password    required      /lib/security/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/pam_unix.so nullok use_authtok md5 shadow
password    sufficient    /lib/security/pam_ldap.so use_authtok
password    required      /lib/security/pam_deny.so

session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
session     optional      /lib/security/pam_ldap.so
```

[13] authconfig is a RedHat utility to configure you pam and nss modules

Warning: a special attention must be taken about the account sufficient parameters as it seems RedHat authconfig tools place it as 'required' in any case (which is not the way you'll need).

5.2 /etc/ldap.conf

edit your /etc/ldap.conf to configure your LDAP parameters :

```
# /etc/ldap.conf for using local LDAP server for authentification

# Your LDAP server. Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base dc=IDEALX,dc=ORG

# RFC2307bis naming contexts
# we use ?sub (and not the default ?one) because we
# separated sambaAccounts on ou=Computers,dc=IDEALX,dc=org
# and ou=Users,dc=IDEALX,dc=org
nss_base_passwd         dc=IDEALX,dc=ORG?sub
nss_base_shadow         dc=IDEALX,dc=ORG?sub
nss_base_group          ou=Groups,dc=IDEALX,dc=ORG?one

ssl no
pam_password md5

# - The End
```

5.3 Test your system

To test your system, we'll create a system account in LDAP (say 'testuser'), and will try login as this new user.

To create an system account in LDAP, use the `smbldap-tool` named smbldap-useradd.pl[14] (assuming you have already configured your `smbldap-tools`):

```
[root@pdc-srv tmp]# smbldap-useradd.pl -m testuser1
adding new entry "uid=testuser1,ou=Users,dc=IDEALX,dc=ORG"

[root@pdc-srv tmp]# smbldap-passwd.pl testuser1
Changing password for testuser1
New password for user testuser1:
Retype new password for user testuser1:
all authentication tokens updated successfully
```

Then, try to login on your system (Unix login) as testuser1 (using another console, or using ssh). Everything should work fine :

[14] see 8 on page 20 for more info

```
[user@host-one:~]$ ssh testuser1@pdc-srv
testuser1@pdc-srv's password:
Last login: Sun Dec 23 15:49:40 2001 from host-one

[testuser1@pdc-srv testuser1]$ id
uid=1000(testuser1) gid=100(users) groupes=100(users)
```

Dont forget to delete this testuser1 after having completed your tests :

```
[root@pdc-srv]# smbldap-userdel.pl testuser1
```

6 Configuring Samba

Here, we'll configure Samba as a Primary Domain Controler for the Microsoft Windows NT Domain named IDEALX-NT with the SAM database stored in our OpenLDAP server.

6.1 Configuration

We need to configure /etc/samba/smb.conf like in the example of 22.4 on page 55, assuming that :

- Our Microsoft Windows NT Domain Name will be : IDEALX-NT
- Our server Netbios Name will be : PDC-SRV
- Our server will allow roving/roaming profiles
- All samba share will rely on /opt/samba/* excepted for home directories (always on /home/USERNAME).
- We really want our Samba-LDAP PDC server to be the domain browser on the LAN.

Edit your /etc/samba/smb.conf like in the example of 22.4 on page 55 to configure your Samba server. Let make some remarques about this file:

the global section This section allow you to configure the global parameter of the server. Here takes places all the parameters we defined in the previous paragraph. We also have defined the program used for a user to change his password (*passwd program*) and the dialog used between the server and the user during the change.

The option "add user script" allow smbd to add, as root, a new machine. When a machine contact the domain, this script is called and the new machine is added to the domain. This makes easily the administration of machine's account. For security, not all the machines could logged to the domain, but only a administrator who has a privilege account.

For french users, we added a line that allow smbd to map incoming filenames from a DOS code page. This option is very useful if you want that files and directories in your profiles are saved with all the accents they have. Don't forget to read the man page for more detail: this option is a Western European UNIX character set. The parameter client code page MUST be set to code page 850 in order for the conversion to the UNIX character set to be done correctly.

```
[global]
  workgroup = IDEALX-NT
  netbios name = PDC-SRV
  server string = SAMBA-LDAP PDC Server
  ...
  passwd program = /usr/local/sbin/smbldap-passwd.pl -o %u
  passwd chat = *new*password* %n\n *new*password* %n\n *successfully*
  unix password sync = Yes
  ...
; SAMBA-LDAP declarations
  ldap suffix = dc=IDEALX,dc=ORG
  ldap admin dn = cn=Manager,dc=IDEALX,dc=ORG
  ldap port = 389
  ldap server = 127.0.0.1
  ldap ssl = No

  add user script = /usr/local/sbin/smbldap-useradd.pl -m -d /dev/null -g 1000 -s /bin/fal
  ...
  character set = iso8859-1
```

the shares sections Here takes place all the share sections. In particular, we can define all the user's home directories which are defined by the [homes] section:

```
[homes]
  comment = Home Directories
  valid users = %S
  read only = No
  create mask = 0664
  directory mask = 0775
  browseable = No
```

Here is the path to the profiles's directory. Profile of all users will be stored here. This is the root directory for profiles and the ldap variable *profilePath* specify exactly the path for each users. For example if the *profilePath* is set to \\PDC-SRV\profiles\testuser, than the profile directory for user *testuser* is /opt/samba/profiles/testuser/. Make sure to have the right permission for this directory. The sticky bit must be set. Make a simple chmod 1757 /opt/samba/profiles and it will be ok. Don't forget that the system doesn't take this change immediately. You should wait several minutes before any profile takes place.

```
[profiles]
  path = /opt/samba/profiles
  writeable = yes
  browseable = no
  create mode = 0644
  directory mode = 0755
  guest ok = yes
```

If you want command's file to be downloaded and ran when a user successfully logged, you have to define a *netlogon* section and a *netlogon script*. The *netlogon script* must take place in the *global* section and the script must be a relative path to the [netlogon] service. For example, if the [netlogon] service specifies a path of */opt/samba/netlogon* (like in our example), than if the script is defined as *logon script = STARTUP.BAT*, then the file that will be downloaded is */opt/samba/netlogon/STARTUP.BAT*. Finally, we defined a *doc* section that authorized everybody to browse the */usr/share/doc* documentation directory.

```
[global]
  ...
  logon script = STARTUP.BAT
  ...

[netlogon]
  comment = Network Logon Service
  path = /opt/samba/netlogon
  guest ok = Yes

[doc]
  path=/usr/share/doc
  public=yes
  writable=no
  read only=no
  create mask = 0750
  guest ok = Yes
```

For example, we could have the STARTUP.BAT script that set the documentation directory mounted on the J volume on windows clients. Another useful command set windows time synchronized to the server's one:

```
NET USE J: \\PDC-SRV\doc
NET TIME \\PDC-SRV /SET /YES
```

6.2 Preparation

You must create some directories, according to your /etc/smb.conf :

```
mkdir /opt/samba
mkdir /opt/samba/netlogon
mkdir /opt/samba/profiles
chmod 1757 /opt/samba/profiles
```

6.3 Initial entries

`Samba` must know the passwd of the `ldap admin dn` (cn=Manager,dc=IDEALX,dc=ORG) you've specified in `smb.conf` to be able to add/modify accounts stored in the LDAP SAM.

To do so, use the following command (assuming 'secret' is the ldap admin dn password, see your /etc/openldap/slapd.conf configuration file to be sure) :

```
[root@pdc-srv samba]# smbpasswd -w secret
Setting stored password for "cn=Manager,dc=IDEALX,dc=ORG" in secrets.tdb
```

Samba will store this datas in /etc/samba/secrets.tbd.

Note that this ldap admin dn may be another account than Root DN : you should use another ldap account who should have permissions to write any sambaAccount and some posixAccount attrs (see **??** on page ??). In this HOWTO, we're using the Root DN.

Then, you should create your 'Administrator' user :

```
[root@pdc-srv samba]# smbldap-useradd.pl -a -m -g 200 administrator
adding new entry "uid=administrator,ou=Users,dc=IDEALX,dc=ORG"

modifying entry "uid=administrator,ou=Users,dc=IDEALX,dc=ORG"

modifying entry "uid=administrator,ou=Users,dc=IDEALX,dc=ORG"


[root@pdc-srv samba]# smbldap-passwd.pl administrator
Changing password for administrator
New password :
Retype new password :
all authentication tokens updated successfully
```

In fact, any user placed in the "Domain Admins" group will be granted Windows admin rights.

6.4 Testing

To validate your Samba configuration, use testparm who should return 'Loaded services file OK.' without any warnings nor unknow parameter. See man testparm for more info.

7 Start-Stop servers

Assuming you're following this HOWTO, we use :

- `OpenLDAP` RedHat 7.2 package,
- `Samba` RedHat RawHide package,
- `nscd` RedHat 7.2 package.

So, to :

- start/stop the `OpenLDAP` server : /etc/init.d/ldap start/stop
- start/stop the `Samba` server : /etc/init.d/smb start/stop
- start/stop the `nscd` server : /etc/init.d/nscd start/stop

8 User management

To manager user accounts, you can use:

1. smbldap-tools, using the following scripts:
   - smbldap-useradd.pl : to add a new user
   - smbldap-userdel.pl : to delete an existing user
   - smbldap-usermod.pl : to modify an existing user data
2. idxldapaccounts if you are looking for a nice Graphical User Interface.

Both method will be presented hereafter.

8.1 A LDAP view

First, let's have a look on what is really a user accounts for LDAP. In fact, there is two kinds of user accounts :

- Posix Accounts, for use with LDAP-aware systems like Unix (*Linux* using pam_ldap and nss_ldap, in this HOWTO). Those kind of accounts use the posixAccount, or shadowAccount if you are using shadow passwords.
- Samba Accounts, for the use of Samba Windows user accounts (and computer accounts too). Those kind of accounts use the sambaAccount LDAP object class (according to the Samba samba.schema).

Here's a LDAP view of an Unix Account (posixAccount in fact, for this HOWTO) :

```
dn: uid=testuser1,ou=Users,dc=IDEALX,dc=ORG
objectClass: top
objectClass: account
objectClass: posixAccount
cn: testuser1
uid: testuser1
uidNumber: 1000
gidNumber: 100
homeDirectory: /home/testuser1
loginShell: /bin/bash
gecos: User
description: User
userPassword: {SSHA}ZSPozTWYsy3addr9yRbqx8q5K+J24pKz

```

FIXME: present a posixAccount (warning : `smbldap-tools v 0.7` will only deal with posixAccount. shadowAccount will be dealed later).

Here's a LDAP view of a Samba user account (sambaAccount) :

```
dn: uid=testsmbuser2,ou=Users,dc=IDEALX,dc=ORG
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: sambaAccount
cn: testsmbuser2
uid: testsmbuser2
uidNumber: 1006
gidNumber: 100
loginShell: /bin/bash
gecos: user-test-2
description: user-test-2
pwdLastSet: 0
logonTime: 0
logoffTime: 2147483647
kickoffTime: 2147483647
pwdCanChange: 0
pwdMustChange: 2147483647
displayName: user-test-2
acctFlags: [UX         ]
rid: 3ee
primaryGroupID: 64
smbHome: \\PDC-SRV\homes
scriptPath: scripts.cmd
lmPassword: 17B4D4AEABF1D7A4AAD3B435B51404EE
ntPassword: 51831BDA51454AECB5D924D0DD12DF8F
userPassword: {SSHA}MhVyay/iN3mxD4y9ELVVQAMT55mu2F0a
homeDirectory: /home/testsmbuser2
homeDrive: J:
profilePath: \\PDC-SRV\profiles\testsmbuser2
```

TODO: explain the LDIF, present attribute types (from schema) and explain them. Here follow a kick explanation about the attributes used:

8.1.1 uid/rid

Samba uses the following calculations:

$\text{userrid} = 2 \times \text{uidNumber} + 1000$ $\text{grouprid} = 2 \times \text{gidNumber} + 1001$

excepted for well-known user rids.

As of Samba 2.2.4, the following holds true:

- the only well-known user rids are `DOMAIN_USER_RID_ADMIN` (0x1F4) and `DOMAIN_USER_RID_GUEST` (0x1F5);
- user and group rids must be given in hexadecimal in LDAP.

However, the rids were written in decimal in LDAP. So at least 2.2.3-pre, Samba do not read them as hexadecimal anymore. The default behaviour of smbldap-useradd.pl as of 20011218 is to use the above calculations and store the rids in decimal.

8.1.2 acctFlags

TODO : explain acctFlags and their usage.

Attribute	**from schema**	**Usage**
cn	core	usually, the username
uid	core	username
description	core	TODO
userPassword	core	password for Unix systems using NSS/PAM LDAP
displayName	inetorgperson	TODO
uidNumber	nis	the numeric user number (Unix and Samba)
gidNumber	nis	the primary group number of the user (Unix)
loginShell	nis	the logon shell used on Unix systems
gecos	nis	the long form of the username
homeDirectory	nis	home directory path for Unix systems
pwdLastSet	samba	The integer time in seconds since 1970 when the lm and ntpasswd were last set.
logonTime	samba	Integer value currently unused
logoffTime	samba	Integer value currently unused
pwdCanChange	samba	Integer value currently unused
pwdMustChange	samba	Integer value currently unused
acctFlags	samba	specify the type of the samba account (W=workstation, U=user, D=disabled, X=no password expiration,...)
rid	samba	the relative identifier (RID) of the user
primaryGroupID	samba	the relative identifier (RID) of the primary group of the user
smbHome	samba	specifies the path of the home directory for the user. The string can be null. If homeDrive is set and specifies a drive letter, homeDirectory should be a UNC path. The path must be a network UNC path. This value can be a null string
scriptPath	samba	The scriptPath property specifies the path of the user's logon script, .CMD, .EXE, or .BAT file. The string can be null. The path is relative to the netlogon share
lmPassword	samba	the LANMAN password
ntPassword	samba	the NT password (md4 hash)
homeDrive	samba	specifies the drive letter to which to map the UNC path specified by homeDirectory. The drive letter must be specified in the form ”driveletter:” where driveletter is the letter of the drive to map. For example: ”Z:”
profilePath	samba	specifies a path to the user's profile. This value can be a null string, a local absolute path, or a UNC path

Table 1: Attributes used for a user Account

8.1.3 scriptPath

The script path override the 'logon script' directive of smb.conf (if exist). Variable substitution (given in this attribute is relative to the netlogon share.

8.2 smbldap-tools

To manipulate user accounts, we've developped a collection of PERL scripts named `smbldap-tools` : they provide all the tools you need to manage user and groups accounts, in a LDAP directory.

Because we've merged posixAccount (and soon, shadowAccount too) and sambaAccount, those scripts may be used to manage Unix and Windows (`Samba`) accounts. As most of existing software are LDAP aware, you can use your SAMBA-LDAP PDC to be an unique source of authentification, and the `smbldap-tools` may offer you a good base to manage user accounts datas.

In this Howto, we have used the following tools to manage user accounts :

- smbldap-useradd.pl : to add an user account (by default a posixAccount. Using '-a' option for a sambaAccount, '-w' option for a machine sambaAccount),
- smbldap-userdel.pl : to delete an existing user account
- smbldap-usermod.pl : to modify an user account.

8.2.1 Create a Unix (Posix) user account

For example, to create a new posixAccount (only usefull for Unix) named testposixuser (we'll use 'coucou' as the password when asked):

```
[root@pdc-srv testsmbuser2]# smbldap-useradd.pl -m testposixuser
adding new entry "uid=testposixuser,ou=Users,dc=IDEALX,dc=ORG"

[root@pdc-srv testsmbuser2]# smbldap-passwd.pl testposixuser
Changing password for testposixuser
New password for user testposixuser:
Retype new password for user testposixuser:
all authentication tokens updated successfully
```

8.2.2 Create an `Samba` user account

For example, to create a new sambaAccount (for use under Unix and `Samba`) named jdoo (we'll use 'coucou' as the password when asked) :

```
[root@pdc-srv testsmbuser2]# smbldap-useradd.pl -a -m -c "John Doo" jdoo
adding new entry "uid=jdoo,ou=Users,dc=IDEALX,dc=org"

modifying entry "uid=jdoo,ou=Users,dc=IDEALX,dc=org"

modifying entry "uid=jdoo,ou=Users,dc=IDEALX,dc=org"

[root@pdc-srv testsmbuser2]# smbldap-passwd.pl jdoo
Changing password for jdoo
New password for user jdoo:
Retype new password for user jdoo:
all authentication tokens updated successfully
```

8.2.3 Setup an user password

You can use smbldap-passwd.pl as a replacement for the system command passwd and the Samba command smbpasswd:

```
[root@pdc-srv testsmbuser2]# smbldap-passwd.pl jdoo
Changing password for jdoo
New password for user jdoo:
Retype new password for user jdoo:
all authentication tokens updated successfully
```

8.2.4 Delete a Posix user account

Just use the following smbldap-tools command:

```
[root@pdc-srv testsmbuser2]# smbldap-userdel.pl -r jdoo
```

In this example, we wanted to remove the user named 'jdoo' and his home directory.

8.2.5 Delete a Samba user account

Exactly like for the deletion of an Unix account, just use smbldap-userdel.pl.

8.2.6 Modify an user account

TODO.

8.3 idxldapaccounts

If you prefer nice GUI to shell, you should have a look on the idxldapaccounts Webmin module. See `http://webmin.idealx.com/`.

TODO: write documentation for these tools

9 Group management

In Samba branch 2_2, only 2 groups are dealed for Microsoft Windows workstations: **Domain Admins** and **Domain Users**. All other groups are considered *Local Unix Group*. That's mean that a Samba user will only be Domain user or Domain Admin. If you only use Samba servers, there will be no problem, but if you plan to use Microsoft Windows NT member server using groups, just forget about it...

To manager group accounts, you can use:

1. smbldap-tools using the following scripts:
   - smbldap-groupadd.pl : to add a new group
   - smbldap-groupdel.pl : to delete an existing group
   - smbldap-groupmod.pl : to modify an existing group
2. idxldapaccounts if you are looking for a nice Graphical User Interface.

Both method will be presented hereafter.

9.1 A LDAP view

First, let's have a look on what is really a user accounts for LDAP. Here's a LDAP view of an user group (for Samba and Unix as it seems that there is no difference for branch 2_2 of Samba):

```
dn: cn=Domain Users,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 201
cn: Domain Users
description: Windows Domain Users
memberUid: testsmbuser2
memberUid: testsmbuser1
```

TODO : explain the LDIF, present attribute types (from schema) and explain them.

9.2 Windows specials groups

The Windows world come with some built-ins users groups :

- FIXME to write (name_of_group : purpose)

TODO: explain the different users groups on Windows/Samba (Domain Admins...).

9.3 smbldap-tools

To manipulate groups, we've developped a collection of PERL scripts named `smbldap-tools` : they provide all the tools you need to manage user and groups accounts, in a LDAP directory.

Because `Samba` use posixGroup, those scripts may be used to manage Unix and Windows (`Samba`) accounts. As most of existing software are LDAP aware, you can use your SAMBA-LDAP PDC to be an unique source of authentification, and the `smbldap-tools` may offer you a good base to manage user accounts datas.

In this Howto, we have used the following tools to manage groups :

- smbldap-groupadd.pl : to add a new group,
- smbldap-userdel.pl : to delete an existing group,
- smbldap-usermod.pl : to modify any group datas (mostly to add or remove an user from a given group).

TODO: write this piece of doc. Show how to manager user and group affectation (removing 1 user from 1 group without too much manipulation when 1000 groups...).

9.4 idxldapaccounts

If you prefer nice GUI to shell, you should have a look on the idxldapaccounts Webmin module. See `http://webmin.idealx.com/`.

TODO: write documentation for these tools

10 Computer management

To manage computer accounts, we'll use the following scripts (from `smbldap-tools`) :

- smbldap-useradd.pl : to add a new computer
- smbldap-userdel.pl : to delete an existing computer
- smbldap-usermod.pl : to modify an existing computer data

Computer accounts are sambaAccounts objects, just like `Samba` user accounts are.

10.1 A LDAP view

Here's a LDAP view of a `Samba` computer account :

```
dn: uid=testhost3$,ou=Computers,dc=IDEALX,dc=ORG
objectClass: top
objectClass: posixAccount
objectClass: sambaAccount
cn: testhost3$
gidNumber: 100
homeDirectory: /dev/null
loginShell: /bin/false
uid: testhost3$
uidNumber: 1005
pwdLastSet: 0
logonTime: 0
logoffTime: 2147483647
kickoffTime: 2147483647
pwdCanChange: 0
pwdMustChange: 2147483647
smbHome: \\%N\nobody
profilePath: \\%N\nobody\profile
description: Computer
rid: 0
primaryGroupID: 0
lmPassword: 7582BF7F733351347D485E46C8E6306E
ntPassword: 7582BF7F733351347D485E46C8E6306E
acctFlags: [W          ]

```

TODO: explain the LDIF, present attribute types (from schema) and explain them.

10.2 Tools

To manipulate computer accounts, we've developped a collection of PERL scripts named `smbldap-tools`: they provide all the tools you need to manage user and groups accounts, in a LDAP directory.

In this Howto, we have used the following tools to manage user accounts :

- smbldap-useradd.pl : to add an computer account, using -w option,

- smbldap-userdel.pl : to delete an existing computer account (FIXME),
- smbldap-usermod.pl : to modify an computer account (FIXME).

TODO: a note on Computer types (W: workstations, S: servers)

TODO: a note on ipHost and other nodes/hosts management system... possible links with DNS/DHCP hosts management (I mean the may be some interaction and we must take care to make all thing works together. see Bind 9 ldap back-end and proposed schema)

10.3 Create a Computer account

To create a computer account, you can use `smbldap-tools` to manually add accounts :

```
[root@pdc-srv root]# smbldap-useradd.pl -w testcomputer1
modifying entry "uid=testcomputer1$,ou=Computers,dc=IDEALX,dc=ORG"
```

You can also use the automatic procedure within you Microsoft Windows client (see your client chapter: Microsoft Windows NT, w2k...) for more information.

10.4 Delete a Computer account

To delete a computer account, just use `smbldap-tools` :

```
[root@pdc-srv root]# smbldap-userdel.pl testcomputer1
```

Instead of removing the computer account, you may want to de-activate the Samba Account. To do that, use an LDAP browser and modify the 'acctFlags' from [W] to [WD] ('D' indicating 'Disabled'). To re-activate the computer account, just modifiy [WD] to [W]. Sometimes, de/re-activation is a better mean to temporary disable the workstation for some times.

11 Profile management

WARNING : Under writing !

TODO: Howto manage profiles (NT profiles, as Unix do the job since... AT&T time...)

11.1 Roaming/Roving profiles

When a Microsoft Windows NT user joined the IDEALX-NT domain, his profile is stored in the directory defined in the *profile* section of the samba configuration file. He has to log out for this to be saved. This is a roaming profile: he can use this profile from any computer he want. If his personal configuration changed, it will be integrated in his roaming profile.

In this Howto, we used roaming profiles: the LDAP ProfilePath indicate to Samba where to look for those roaming profile (

PDC-SRV
profiles
testsmbuser2, and the [profiles] section of the /etc/samba/smb.conf indicate to samba how to deal with those profiles.

Keep in mind that a 'regular' roaming profile is about 186 Kb of data (even more if users uses big GIF or BMP image as background picture ...): don't forget impact on load/traffic...

11.2 Mandatory profiles

The mandatory profile is created by the same way of the roaming profile. The difference is that his profile is made read only by the administrator so that the user can have only one fixed profile on the domain.

To do so, rename the file NTuser.dat to NTuser.man (for MANdatory profile), and remove the right access bit. For our *testsmbuser1* user, you'll have to do:

```
mv /opt/samba/profiles/testsmbuser1/NTUSER.DAT /opt/samba/profiles/testsmbuser1/NTUSER.MAN
chmod -w /opt/samba/profiles/testsmbuser1/NTUSER.MAN
```

This way, you may want to set up a common user profile for every user on the Domain.

11.3 Logon Scripts

To use Logon Scripts (.BAT or .CMD), just specify the relative path from the netlogon share to the command script desired in the **scriptPath** attribute for the user.

Variable substitutions (the logon script smb.conf directive when you're using LDAP.

11.4 LDAP or not LDAP?

Perhaps, you'll want to use an alternative system policy concerning profiles : granting some user the roaming profile privilege across the domain, while some other may have only roaming profile on one PDC server, and some other won't use roaming profile at all. This alternative way is possible thanks to `Samba` who will search in the LDAP sambaAccount for the profile location if no information is given by the 'logon drive', 'logon script' and 'logon path' directives of smb.conf.

We'll discuss this alternative in a future revision of this document.

12 Workstations integration

12.1 Microsoft Windows 95 and 98

TODO

12.2 Microsoft Windows NT

TODO

12.3 Microsoft Windows 2000 and XP

TODO: use the W2K requester, using a domain admin group member account.

NICE: screenshots.

12.3.1 RequireSignOrSeal

This registry key (gathered from the Samba-tng lists) is needed for Windows 2000 and XP clients to join and logon to a Samba domain :

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\netlogon\parameters
"RequireSignOrSeal"=dword:00000000
```

You can change this in the Local or Domain policy editor in Windows 2000.

12.3.2 Fake user root

To allow Microsoft Windows 2000 and XP workstatin to join the domain, a root user must exit (uid=0) and be used when joining a client to the domain [15].

To create this false user (false because the user root should be present on you're system files, not in LDAP), just issue the following commands:

```
smbldap-useradd.pl -a -m -g 200 root
smbldap-usermod.pl -u 0 -g 0 root
smbldap-passwd.pl root
```

This workaround permit to avoit the creation of this fake user root, but permit a massive security hole if used as Samba have no real access control on passdb backends :

[15] a workaround/patch exist but will permit a massive security hole if used

```
--- passdb/pdb_ldap.c.orig      Thu May 16 00:17:39 2002
+++ passdb/pdb_ldap.c   Thu May 16 00:20:36 2002
@@ -75,11 +75,16 @@ static BOOL ldap_open_connection (LDAP *
        int version, rc;
        int tls = LDAP_OPT_X_TLS_HARD;

+/* Q&D patch : permit non root bind to LDAP
+   because if so (original code), you cannot add W2K/WXP workstations accounts
+   via the W2K/WXP requester, using an uid != from 0 (ex: user 'administrator'
+   from a " @"Domain Admin" " group (from 'domain admin group' directive in smb.conf)
+
        if (geteuid() != 0) {
                DEBUG(0, ("ldap_open_connection: cannot access LDAP when not root..\n"));
                return False;
        }
-
+*/
        if (lp_ldap_ssl() == LDAP_SSL_ON && lp_ldap_port() == 389) {
                port = 636;
        }
```

12.4 Linux and Unix

TODO

13 Servers integration

13.1 Samba Member Server

TODO: explain configuration

The smb.conf of this Samba member server should indicate:

```
; Samba Domain Member server
; like the Samba-LDAP PDC but without security user and LDAP directives, but
; the followin lines:
security        domain
password server         =        hostname.fqdn (or IP address) of the Samba-LDAP PDC
; note: this samba server does not need to be compiled with
; --with-ldapsam option
```

Once configured and started, you should add the machine account on the PDC, using the following commands:

```
root@on-the-PDC# smbldap-useradd -w short-hostname-of-the-samba-member-server
```

and then, on the Samba member server itself:

```
root@on-the-member-server# smbpasswd -j "IDEALX-NT"
```

13.2 Samba BDC Server

TOD0: explain. explain alternatives

13.3 **Microsoft Windows NT** Member Server

TODO: explain

13.4 **Microsoft Windows NT** BDC Server

TODO: explain why not :-)

13.5 **Microsoft Windows 2000** Member Server

TODO: explian

13.6 **Microsoft Windows 2000** BDC Server

TODO: explain why not :-)

14 Tests procedure

The test-list presented in this section are common to all windows system's versions. If one version may cause problem, or if the procedure is different, we'll make a special note.

14.1 Global configuration

This section help you to test the good configuration and the good operation of your samba-ldap system. We suppose that your system is running all the needed services. You can verify this using the following steps :

- If you have problems starting samba, you can use the testparm command to see if the configuration's file syntax is right. You can verify using the following command line :

```
[root@PDC-SRV root]# ps afuxw | grep smb
0         17049  0.0  0.7  5524 1888 ?        S    11:45   0:00 smbd -D
1002      17146  0.0  1.3  7184 3408 ?        S    11:50   0:00  \_ smbd -D
0         17223  0.1  1.2  7060 3140 ?        S    12:00   0:00  \_ smbd -D
[root@PDC-SERV root]# ps afuxw | grep nmb
0         17054  0.0  0.7  4636 1856 ?        S    11:45   0:00 nmbd -D
0         17057  0.0  0.6  4584 1552 ?        S    11:45   0:00  \_ nmbd -D
```

- is your ldap server up ? You can verify using the following command line :

```
[root@PDC-SRV root]# ps afuxw | grep ldap
ldap      12358  0.0  5.0 16004 12972 ?       S    Nov14   0:03 /usr/sbin/slapd -u lda
```

or

```
[root@PDC-SRV root]# netstat -tan | grep LISTEN | grep 389
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN
```

14.2 Adding a new computer in the domain by creating an account manually

If you want the computer named "testmachine" to be added to the domain IDEALX-NT, you must create a account for it. This can be manually done using the script smbldap-useradd.pl previously described in the section **??** on page ??. Then you can add the computer in the domain, following this steps :

for Microsoft Windows NT 4 (SP1, SP6):

- logged into Microsoft Windows NT using the administrator account
- click on the "start" menu, "Parameters" and "Configuration"

- double click on ”Network” and the ”modify” button
- you must now see the machine’s name and the domain’s name. You have to change the default parameters, or modifie a previous configuration. Then select the ”domain” option and add the name of the domain you want to join.
- click on the ”ok” button
- the computer is already registered so that you normally have the welcome message ”welcome to domain IDEALX-NT”
- restart your windows system.

for Microsoft Windows NT 2000:

- logged into windows using the administrator account.
- click on the ”start” menu, ”Parameters” and ”Configuration”.
- double click on ”System”, select the onglet ”Network identification” and then ”properties”.
- you must now see the machine’s name. You have to change the default parameters, or to modifie a previous configuration by indicating the domaine name.
- the computer is already registered so that you normally have the welcome message ”welcome to domain IDEALX-NT”
- restart your windows system.

14.3 Adding a new computer in the domain automatically

A second way to do this can be directly done directly from Microsoft Windows NT environnement, using the administrator priviledged account. This procedure will create automatically an account for the comuter, and will also join it to the domain.

To do so, follow the same steps as the previous section described in section 14.2 on the page before. When informing the domain name, ask for creating a new compuetr account, and add the administrator account For Microsoft Windows NT 2000, the account is asked when prssing the ”ok” button.

- Login : administrator
- Password : coucou

14.4 Creating an user account

You cannot[16] create user accounts with Microsoft Windows NT Domain management tools: you must use the `smbldap-tools` (or any other LDAP manipulation tools). To do so, see section 8 on page 20. If interested in a graphical user interface to manager user and group accounts, please have a look on the `idxldapaccounts` Webmin module available at `http://webmin.idealx.org/`

To test:

- create an user account for 'testsmbuser' (**??** on page ??)
- verify this user account is ok :

  ```
  $id testsmbuser
  ```

  should return something like that:

  ```
  [root@speed3 samba]# id testsmbuser
  uid=1008(testsmbuser) gid=100(users) groups=100(users),201(Domain Users)
  ```

- additionnaly, if you're using an ldapbrowser, you should see the new uid=testsmbuser,ou=Users,dc=IDEA in the directory.

14.5 Logging in the domain as testsmbuser

You need to use an already Domain added workstation to proceed this test. This is previously explained is section 14.2 or 14.3.

Call the Winlogon (CTRL-ALT-SUPPR), and enter:

- Login : testsmbuser
- Password : coucou[17]
- Domain : IDEALX-NT

You should then log on fine. When you log in the domain with your username testsmbuser, verify that those differents points are ok:

- browse your personal folder and all shared folders, and read a file
- create a new file in your home directory, verify that you can save it
- verify that all permissions seems right: you can't browse a directory you don't have the permissions to, you can't edit or/and modify a file you don't have permissions to.

[16] AFAIK with release 2.2.4 of `Samba`

[17] in fact, the one you gave in the section : **??** on page ??

15 Real life considerations

Now we've detail how to set up your brand new PDC-Killer prototype, we're ready to go further: the real life, the one where users don't care about looking for solutions to a given problem, but will first consider they've got one and you're the guilty :-)

To struggle in this pleasant world, you should have a look on the following considerations : they may help you.

First, if this HOWTO was your fist approach with Samba and OpenLDAP, you should have a look on:

- a very good OpenLDAP brief by Adam Williams available at ftp://kalamazoolinux.org/pub/pdf/ldapv3.pdf: an excellent presentation/briefing on OpenLDAP on the *Linux* Platform.
- the OpenLDAP project website,
- the Samba project website,
- numerous documentation (printed or not) done on these two topics (Teach Yourself Samba in 24 hours for example).

15.1 Performance

15.1.1 Lower Log Level in production

When everything is okay with you configuration, you are **strongly encouraged** to lower log levels for better performance.

Best practices are to activate debuging logs only when you want to investigate a potential problem, and stay with low log level (or no log at all if you're seeking maximum performance) during exploitation time (most of the time as Samba really a robust implementation, thank's to the Samba Team).

Here's is an example of a standard exploitation mode log management parameters for a Samba server :

```
  log file = /var/log/samba/%m.log
  log level = 0
  max log size = 5000
```

15.1.2 OpenLDAP tunning

You should consider indices on your directory server. For OpenLDAP, the following should be ok for a PDC like the one we described in this HOWTO:

```
# index
index   objectClass,rid,uid,uidNumber,gidNumber,memberUid      eq
index   cn                                                     eq,subinitial
```

Of course, indices depends on you directory usage. Consult the OpenLDAP documentation for more info.

Have a look on the following slapd.conf directives too:

- loglevel: lower to '0' for production purpose
- lastmod: set it to 'off' if you really don't need it
- cachesize: set a confortable cache size (say 1000 for a mid-level production site for 1000 users),
- dbcachesize: set a confortable db cache size (say 10000 for a mid-level production site for 1000 users)
- dbnosync: in case you're fool enought to think nothing will never crash :-)

15.2 Security

15.2.1 Use an account which is not Root DN

In this HOWTO, we're using the Root DN : the *ldap admin dn* should be another account than Root DN : you should use another ldap account who should have permissions to write any sambaAccount and some posixAccount attrs.

15.2.2 Use SSL!

In this HOWTO, whe are using clear LDAP transport between Samba and OpenLDAP. As both servers implement SSL, you should use LDAPS transport instead.

15.2.3 Use ACLs for LDAP

Place ACLs to protect the directory datas. For the usage of Samba, the following should deliver basic protection:

```
# Password hashes password
access to attrs=userPassword
      by self write
      by anonymous auth
      by * none
access to attrs=lmPassword
      by self write
      by anonymous auth
      by * none
access to attrs=ntPassword
      by self write
      by anonymous auth
      by * none

# Global read access
access to *
      by * read
```

15.3 Backup your datas

TODO: how to backup and restore your PDC !

Crucial ! Some scripts may help do the job (even if not used, the will explain what to backup exactly, and how to restore). In fact, those scripts just have to backup: config files (ldap, nss, ldap, samba and tbds..) and the 'SAM' (so a LDIF may do the job). An smbldap-backup and smbldap-restore?

16 Load and Availability

TODO: indicate some load params, and present a redundant and HA solution.

TODO: describe test-plateform.

16.1 OpenLDAP Load

As we're storing users and groups in a LDAP directory, we will have a closer look on the OpenLDAP capacity to store numerous account, and systems (Samba and pam_ldap) to interact with this LDAP database.

For testing purpose, we're going to test bind/read/write operations on LDAP, with a population of 50.000 users, 50.000 computers. and 1000 groups.

16.2 Samba Load

As we're storing the SAM database in a LDAP directory, we will have a closer look on the Samba-LDAP capacity to interact under heavy stress.

For testing purpose, we're going to compare Samba with and without the LDAP stored SAM.

We'll have to show stress test results (smbtorture?) using 20, 50, 100, 150 and 200 clients.

16.3 High Availability

TODO: Present an HA configuration: what to do, how to do it (using Kimberlite/Mon or Hearbeat/Mon).

17 Migration

In this section, we'll describe how to migrate from a Microsoft Windows NT PDC Server to a Samba+LDAP Domain Controler, in two different user cases:

- migration from a given Domain (the old one) to another (the new one),
- the same Domain is used

In both cases, emphasis must be placed on transparency of migration: movement to the new system (Samba+LDAP) should be accomplished with the absolute minimum of interference to the working habits of users, and preferably without those users even noticing that is has happened, if feasible.

In both cases, migration concern the following informations:

1. users accounts (humans and machines),
2. groups and group members,
3. users logon scripts,
4. users profiles (NTUSER.DAT),
5. all datas,
6. all shares and shares permissions informations,
7. all NTFS ACLs used by users on shares.

17.1 General issues

17.1.1 Users and machines accounts

Dumping the Microsoft Windows NT registry with PWDUMP Users and machine accounts can be extracted from the Microsoft Windows NT SAM database, using the pwdump utility: this handy utility dumps the password database of an NT machine that is held in the NT registry into a valid smbpasswd format file. This utility may be downloaded from `ftp://ftp.samba.org/pub/samba/pwdump/`. We use it instead of the net /domain NT command because we want to retrieve the LANMAN and the NT passwords to left them unchanged during the migration.

This utility must be run as 'Administrator' in the PDC where the SAM to be migrated reside. It dumps NT password entries in the format:

```
<user>:<id>:<lanman pw>:<NT pw>:<comment>:<homedir>:
```

Where:

- ¡user¿ is the user-name on Microsoft Windows NT,
- ¡id¿ is the Microsoft Windows NT RID (Relative ID), the last 32 bits of the Microsoft Windows NT user SID;
- ¡lanman pw¿ is the LANMAN password hash (see below);
- ¡NT pw¿ is the Microsoft Windows NT password hash (md4 in fact). If the user has no password, the entry will be dumped as **NO PASSWORD*****. If the entry is disabled or invalid, these are dumped as 32 '*' characters;
- ¡comment¿ is the concatenation of the user full name on Microsoft Windows NT and the description field in the Microsoft Windows NT user-manager program;
- ¡homedir¿ cannot contain ':' as this character is used as field separators. All ':' characters after drive letter are dumped as '_' .

pwdump dumps users and machine accounts (machine accounts use the '$' character at the end of their name).

Populating the LDAP directory with accounts Using the SAM output, we have to use the **smbldap-migrate-accounts.pl** tool (part of the `smbldap-tools`) to update the LDAP repository (`smbldap-tools` must be correctly configured at this time).

Basically, **smbldap-migrate-accounts.pl** take a 'pwdump' flat file to update the master LDAP repository using the following parameters:

- **-a** : process only people, ignore computers,
- **-w** : process only computers, ignore persons,
- **-A opts**: a string containing arguments to pass verbatim to smbldap-useradd when adding users, eg "-m -x". You don't have to specify -a in this string,
- **-W opts**: a string containing arguments to pass verbatim to smbldap-useradd when adding computers, eg "-m -x". You don't have to specify -w in this string,
- **-C** : if NT account not found in LDAP, don't create it and log it to stdout (default is to create the account),
- **-U** : if NT account found in LDAP, don't update it and log it to stdout (default is to update the account).

For example, if you want to create initial entries to the LDAP repository, and if you think your PDC is the most up to date source of information, just issue the following command :

```
smbldap-migrate-accounts.pl < pwdump-file.txt
```

If you just want to update data from PDC to the LDAP directory, but don't want to create any new accounts (perhaps as they are not all 'regular accounts'), and want to create the home directory, just issue the following command, on the server you are configuring:

```
smbldap-migrate-accounts.pl -C -A "-m" < pwdump-file.txt
```

17.1.2 Groups and members

To be written ! as the tools they are based on (smbldap-migrate-groups.pl, part of the smbldap-tools).

17.1.3 Logon scripts

Logon scripts are DOS scripts that are run every time someone logs on. They must be placed on the **[netlogon]** special share, and you can specify, for each user, the location of this script in the *scriptPath* LDAP attribute.

For example, if you special netlogon share is defined like the following example, in your /opt/samba/etc/smb.conf:

```
[netlogon]
        comment = Network Logon Service
        path = /data/samba/netlogon
        guest ok = Yes

```

And you want the user **myuser** to execute the script named myuser.cmd, just complete the following operations:

- copy the myuser.cmd from the old PDC to the new *Linux* server on /opt/samba/netlogon/myuser.cmd,
- modify the LDAP user definition by placing myuser.cmd on the *scriptPath* attribute,
- logon as **myuser** on a Microsoft Windows NT (or Microsoft Windows 2000) workstation connected to the domain, just to test the logon script activation on login.

So, to migrate all logons scripts from the old Microsoft Windows NT PDC to the new *Linux* server, just copy all logon scripts (placed in C:\WINNT\sysem32\repl\import\) to /opt/samba/netlogon/, and modify your *scriptPath* users definitions in the LDAP directory to record the name of the user's logon scripts.

Note that the old 'logon scripts' directive of smb.conf will no longer be used when using Samba and LDAP together, with release 2.2.4 of Samba.

17.1.4 Users profiles

To be written.

17.1.5 Datas

To be written. Use Rsync !

17.1.6 Shares and permissions

To be written.

17.1.7 NTFS ACLs

To be written. use chacl !

17.2 Same domain

To be written.

17.3 Changing domain

To be written.

18 Contributions

Some useful scripts and tools may help you when setting up your `Samba+OpenLDAP` PDC server:

- `smbldap-tools`: PERL scripts to manager user and group accounts. See `http://samba.idealx.org/`. Note that these scripts are now shipped with `Samba` release 2.2.5,
- `idxldapaccounts` Webmin module: a `Webmin` module to manager user and group accounts in a PDC configuration, via `Webmin` graphical user interface. See `http://webmin.idealx.org`.

19 Thanks

This document is a collective work to :

- quickly discover the LDAP PDC functionnalities of Samba,
- quickly have a working configuration,
- prepare a good update for the SAMBA-PDC-HOWTO :-)

The following people directly worked on this Howto :

- David Le Corfec (dlecorfec@IDEALX.com),
- Jérôme Tournier (jtournier@IDEALX.com),
- Michael Weisbach (mwei@tuts.nu),
- Stefan Schleifer (stefan.schleifer@linbit.com).

The authors would like to thank the following people for providing help with some of the more complicated subjects, for clarifying some of the internal workings of Samba or OpenLDAP, for pointing out errors or mistakes in previous versions of this document, or generally for making suggestions (in alphabetical order):

- Ignacio Coupeau (icoupeau@unav.es),
- Michael Cunningham (archive@xpedite.com),
- Adam Williams (awilliam@whitemice.org),
- Some people on **irc.openproject.org #samba-technical**
- Samba and SAMBA-TNG Teams of course !

20 Frequently Asked Questions... and answers

I can't create a windows account from Microsoft Windows NT 4 itself: try adding it manually, using the script *smbldap-useradd.pl* (you must be root on the PDC server). If your machine's name is VMNT, then the command line is:

```
smbldap-useradd.pl -w VMNT$
```

I can't join the domain: many reason can cause this problem. verify the following points:

- in the samba configuration file (smb.conf), put the *interface* parameter to the interface is listening the network on. We originally put "interfaces = 192.168.2.0/24 127.0.0.1/32" which caused the "can't join the domain" problem.

my profiles are not saved on the server: make sure that the profile directory on the server has the right permissions. You must do a

```
chmod 1757 /opt/samba/profiles}
```

I deleted my computer from the domain, and i can't connect to it anymore: When you leave the domain IDEALX-NT, you have to reboot your machine. If you don't, you will not be able to join any more the domain. If you done this and it still doesn't work, remove the machine's account from the ldap entry and recreate it. For this, use the command

```
TO DO
```

21 Samba-Ldap on Debian Woody

The standard Samba Debian package is compiled with PAM Support. So you have to get the samba source and recompile it yourself.

For this howto, I used Samba version 2.2.4-1:

```
# apt-get source samba
```

Then, in the samba-2.2.4/debian edit the following files:

- rules: get rid of any pam compile options. I have added any missing options mentioned in this redhat howto. Also comment some files which are not created (so don't install or move them):

```
[ -f source/Makefile ] || (cd source && ./configure \
    --host=$(DEB_HOST_GNU_TYPE) \
    --build=$(DEB_BUILD_GNU_TYPE) \
    --with-fhs \
    --prefix=/usr \
    --sysconfdir=/etc \
    --with-privatedir=/etc/samba \
    --localstatedir=/var \
    --with-netatalk \
    --with-smbmount \
    --with-syslog \
    --with-sambabook \
    --with-utmp \
    --with-readline \
    --with-libsmbclient \
    --with-winbind \
    --with-msdfs \
    --with-automount \
    --with-acl-support \
    --with-profile \
    --disable-static \
    --with-ldapsam)


#install -m 0644 source/nsswitch/pam_winbind.so \
    #$(DESTDIR)/lib/security/

#mv $(DESTDIR)/usr/bin/pam_smbpass.so $(DESTDIR)/lib/security/

#cp debian/samba.pamd $(DESTDIR)/etc/pam.d/samba
```

- libpam-smbpass.files: get rid of the lib/security/pam_smbpass.so entry (yes the file is then empty),

- samba-common.conffiles: get rid of the /etc/pam.d/samba entry (yes the file is then empty)
- winbind.files: get rid of the lib/security/pam_winbind.so

Afterwards make a dpkg-buildpackage from the main directory level. when finished you have the .deb files ready to be installed:

```
# dpkg -i samba-common_2.2.4-1_i386.deb libsmbclient_2.2.4-1_i386.deb
samba_2.2.4-1_i386.deb smbclient_2.2.4-1_i386.deb smbfs_2.2.4-1_i386.deb
swat_2.2.4-1_i386.deb winbind_2.2.4-1_i386.deb
```

the global part of a sample smb.conf looks like this:

```
[global]
   workgroup = Test
   netbios name = MARY
   domain admin group = @domadmin
   server string = %h server (Samba %v)
;   wins support = yes  <== important with wins support, it didn’t work for me
   interfaces = 10.1.1.180
   invalid users = root
   log file = /var/log/samba/log.%m
   log level = 1
   max log size = 1000
   syslog = 0
   encrypt passwords = true
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
   local master = yes
   preferred master = yes
   dns proxy = yes
   unix password sync = true
   passwd program = /usr/local/bin/smbldap-passwd.pl -o %u
   passwd chat = *new*password* %n\n *new*password:* %n\n *successfully*
   unix password sync = Yes

# SAMBA-LDAP Declarations
   ldap suffix = dc=domain,dc=com
   ldap admin dn  = cn=admin,dc=domain,dc=com
   ldap port = 389
   ldap server = 10.1.1.15
   ldap ssl = No
   add user script = /usr/local/bin/smbldap-useradd.pl -m -d /dev/null -g 1000 -s /bin/false

```

22 Annexes

Here you’ll find some sample documentations and config files, used in this HOWTO.

22.1 samba.schema

The `Samba` schema is shipped with `Samba`-2.2.4 source code (in example/LDAP/). Please note that this schema is subject to change (probably in 2.2.5, the ’sambaAccount’ objectClass will become AUXILLIARY).

For this HOWTO purpose, we commented the ’displayName’ attributetype, as we’re using inetOrgPerson too (and ’displayName’ is already defined in inetOrgPerson.schema). Here’s the ’patched’ schema we’ve used :

```
##
## schema file for OpenLDAP 2.0.x
## Schema for storing Samba’s smbpasswd file in LDAP
## OIDs are owned by the Samba Team
##
## Prerequisite schemas - uid (cosine.schema)
##                      - displayName (inetorgperson.schema)
##
## 1.3.6.1.4.1.7165.2.1.x - attributetypes
## 1.3.6.1.4.1.7165.2.2.x - objectclasses
##

##
## Password hashes
##
attributetype ( 1.3.6.1.4.1.7165.2.1.1 NAME ’lmPassword’
        DESC ’LanManager Passwd’
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.2 NAME ’ntPassword’
        DESC ’NT Passwd’
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{32} SINGLE-VALUE )

##
## Account flags in string format ([UWDX     ])
##
attributetype ( 1.3.6.1.4.1.7165.2.1.4 NAME ’acctFlags’
        DESC ’Account Flags’
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{16} SINGLE-VALUE )

##
## Password timestamps & policies
##
attributetype ( 1.3.6.1.4.1.7165.2.1.3 NAME ’pwdLastSet’
        DESC ’NT pwdLastSet’
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.5 NAME ’logonTime’
        DESC ’NT logonTime’
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```

attributetype ( 1.3.6.1.4.1.7165.2.1.6 NAME 'logoffTime'
        DESC 'NT logoffTime'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.7 NAME 'kickoffTime'
        DESC 'NT kickoffTime'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.8 NAME 'pwdCanChange'
        DESC 'NT pwdCanChange'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.9 NAME 'pwdMustChange'
        DESC 'NT pwdMustChange'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

##
## string settings
##
attributetype ( 1.3.6.1.4.1.7165.2.1.10 NAME 'homeDrive'
        DESC 'NT homeDrive'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{4} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.11 NAME 'scriptPath'
        DESC 'NT scriptPath'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.12 NAME 'profilePath'
        DESC 'NT profilePath'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.13 NAME 'userWorkstations'
        DESC 'userWorkstations'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{255} SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.17 NAME 'smbHome'
        DESC 'smbHome'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{128} )

attributetype ( 1.3.6.1.4.1.7165.2.1.18 NAME 'domain'
        DESC 'Windows NT domain to which the user belongs'
        EQUALITY caseIgnoreIA5Match
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{128} )

##
## user and group RID
##
attributetype ( 1.3.6.1.4.1.7165.2.1.14 NAME 'rid'
        DESC 'NT rid'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

attributetype ( 1.3.6.1.4.1.7165.2.1.15 NAME 'primaryGroupID'
        DESC 'NT Group RID'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```

##
## The smbPasswordEntry objectclass has been depreciated in favor of the
## sambaAccount objectclass
##
#objectclass ( 1.3.6.1.4.1.7165.2.2.1 NAME 'smbPasswordEntry' SUP top AUXILIARY
#        DESC 'Samba smbpasswd entry'
#        MUST ( uid $ uidNumber )
#        MAY  ( lmPassword $ ntPassword $ pwdLastSet $ acctFlags ))

objectclass ( 1.3.6.1.4.1.7165.2.2.2 NAME 'sambaAccount' SUP top STRUCTURAL
        DESC 'Samba Account'
        MUST ( uid $ rid )
        MAY  ( cn $ lmPassword $ ntPassword $ pwdLastSet $ logonTime $
               logoffTime $ kickoffTime $ pwdCanChange $ pwdMustChange $ acctFlags $
               displayName $ smbHome $ homeDrive $ scriptPath $ profilePath $
               description $ userWorkstations $ primaryGroupID $ domain ))

##
## Used for Winbind experimentation
##
objectclass ( 1.3.6.1.4.1.7165.1.2.2.3 NAME 'uidPool' SUP top AUXILIARY
        DESC 'Pool for allocating UNIX uids'
        MUST ( uidNumber $ cn ) )

objectclass ( 1.3.6.1.4.1.7165.1.2.2.4 NAME 'gidPool' SUP top AUXILIARY
        DESC 'Pool for allocating UNIX gids'
        MUST ( gidNumber $ cn ) )
```

22.2 base.ldif

Here's a LDIF output of initial entries for the OpenLDAP server. Most of the groups are not of any usage (excepting being groups, which is afterall enought to be usable :-).

In this HOWTO, we used the 'Domain Users' group to be the default group all Samba users belong. The user 'nobody' is member of the 'Guests' group.

```
dn: dc=IDEALX,dc=ORG
objectClass: domain
dc: IDEALX

dn: ou=Groups,dc=IDEALX,dc=ORG
objectClass: top
objectClass: organizationalUnit
ou: Groups
description: System Groups

dn: ou=Users,dc=IDEALX,dc=ORG
objectClass: top
objectClass: organizationalUnit
ou: Users
description: Users of the Organization

dn: ou=Computers,dc=IDEALX,dc=ORG
objectClass: top
objectClass: organizationalUnit
ou: Computers
description: Windows Domain Computers

dn: cn=Domain Admins,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 200
```

```
cn: Domain Admins
memberUid: administrator
description: Windows Domain Users

dn: cn=Domain Users,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 201
cn: Domain Users
description: Windows Domain Users

dn: cn=Domain Guests,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 202
cn: Domain Guests
description: Windows Domain Guests Users

dn: cn=Administrators,ou=Groups,dc=IDEALX,dc=ORG
description: Members can fully administer the computer/domain
objectClass: posixGroup
gidNumber: 220
cn: Administrators
description: Windows Domain Members can fully administer the computer/domain

dn: cn=Users,ou=Groups,dc=IDEALX,dc=ORG
description: Ordinary users
objectClass: posixGroup
gidNumber: 221
cn: Users
description: Windows Domain Ordinary users

dn: cn=Guests,ou=Groups,dc=IDEALX,dc=ORG
description: Users granted guest access to the computer/domain
objectClass: posixGroup
gidNumber: 222
cn: Guests
memberUid: nobody
description: Windows Domain Users granted guest access to the computer/domain

dn: cn=Power Users,ou=Groups,dc=IDEALX,dc=ORG
description: Members can share directories and printers
objectClass: posixGroup
gidNumber: 223
cn: Power Users
description: Windows Domain Members can share directories and printers

dn: cn=Account Operators,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 224
cn: Account Operators
description: Windows Domain Users to manipulate users accounts

dn: cn=Server Operators,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 225
cn: Server Operators
description: Windows Domain Server Operators

dn: cn=Print Operators,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 226
cn: Print Operators
description: Windows Domain Print Operators

dn: cn=Backup Operators,ou=Groups,dc=IDEALX,dc=ORG
objectClass: posixGroup
gidNumber: 227
```

```
cn: Backup Operators
description: Windows Domain Members can bypass file security to back up files

dn: cn=Replicator,ou=Groups,dc=IDEALX,dc=ORG
description: Supports file replication in a domain
objectClass: posixGroup
gidNumber: 228
cn: Replicator
description: Windows Domain Supports file replication in a domain
```

22.3 /etc/ldap.conf

Here's an complete sample /etc/ldap.conf used in this HOWTO.

```
# $Id: ldap-conf.tex,v 1.3 2002/06/06 05:31:00 olem Exp $
# $Source: /cvs/public/samba/samba-ldap-howto/ldap-conf.tex,v $
#
# /etc/ldap.conf for using local LDAP server for authentification

# Your LDAP server. Must be resolvable without using LDAP.
host 127.0.0.1

# The distinguished name of the search base.
base dc=IDEALX,dc=org

# RFC2307bis naming contexts
nss_base_passwd                dc=IDEALX,dc=org?sub
nss_base_shadow         dc=IDEALX,dc=org?sub
nss_base_group          ou=Groups,dc=IDEALX,dc=org?one

ssl no
pam_password md5

# - The End
```

22.4 smb.conf

Here's an sample /etc/samba/smb.conf used in this HOWTO.

```
[global]
  workgroup = IDEALX-NT
  netbios name = PDC-SRV
  server string = SAMBA-LDAP PDC Server
  encrypt passwords = Yes
  passwd program = /usr/local/sbin/smbldap-passwd.pl -o %u
  passwd chat = *new*password* %n\n *new*password* %n\n *successfully*
  unix password sync = Yes

  log file = /var/log/samba/%m.log
  log level = 5 ; remember to lower the log level in real life :-)
  max log size = 0


  socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

  domain logons = Yes
  os level = 80
  preferred master = False
  domain master = True
  dns proxy = No
  wins support = Yes

  ; SAMBA-LDAP declarations
  ldap suffix = dc=IDEALX,dc=ORG
  ldap admin dn = cn=Manager,dc=IDEALX,dc=ORG
  ldap port = 389
  ldap server = 127.0.0.1
  ldap ssl = No

  printing = lprng

  ; Deactivate opportunistic locks (wised)
  ; opLocks = False
  ; encoding to french
  character set = iso8859-1

  ; using smbldap-tools to add machines
  add user script = /usr/local/sbin/smbldap-useradd.pl -w %u
  ; users and groups allowed to be ’Domain Admins’
  domain admin group = " @"Domain Admins" "

[homes]
  comment = Home Directories
  valid users = %S
  read only = No
  create mask = 0664
  directory mask = 0775
  browseable = No

[netlogon]
  comment = Network Logon Service
  path = /opt/samba/netlogon
  guest ok = Yes

[profiles]
  path = /opt/samba/profiles
  writeable = yes
  browseable = no
  create mode = 0644
  directory mode = 0755
  guest ok = yes

[printers]
  comment = All Printers
  path = /var/spool/samba
  printable = Yes
  browseable = No

[tmp]
  comment = Temporary file space
  path = /tmp
  read only = No
  guest ok = Yes



```