By Richard
Published: 2008-01-04 15:39

OpenLDAP + Samba Domain Controller On Ubuntu 7.10Preface

This document is a step by step guide for configuring Ubuntu 7.10 as a Samba Domain Controller with an LDAP backend (OpenLDAP). The point is to configure a server that can be comparable, from a central authentication point of view, to a Windows Server 2003 Domain Controller. The end result will be a server with an LDAP directory for storing user, group, and computer accounts. A Windows XP Professional SP2 workstation will be able to join the domain once properly configured. Please note that you do not have a fully comparable Windows domain controller at this time. Do not kid yourself, this guide only gets you a server with LDAP authentication. Of course this can be expanded to include slave servers to spread out authentication over multiple networks. Please also note that it took me approximately two and a half weeks to compile this information and get it working. The same functionality can be had in Windows in less than four hours (and this includes operating system installation). In my humble opinion the open source community will need to work on this side of Linux in order for it to be a true alternative to Windows.

Legal/Warranty/Etc...

This document is provided as-is with no implied warranty or agreement. I will not support other systems without compensation. This document is the property of Richard Maloley II. This document may be redistributed, copied, printed, and modified at will, however my name must remain as the original source. Legal action can and will be brought against any and all infractions of the terms.

Special Items of Interest

* My hostname during the installation was set to: *dc01-ubuntu*
 * My fully qualified domain name will be: *dc01-ubuntu.example.local*
 * After the installation my */etc/hostname* was changed to: *dc01-ubuntu.example.local*
 * After the installation my */etc/hosts* was changed so that the line 127.0.1.1 contained "dc01-ubuntu dc01-ubuntu.example.local" to ensure no issues with name resolution.
 * My LDAP domain is: *example.local*
 * This translates to a Base DN of: *dc=example,dc=local*
 * All passwords used are "12345" to keep things simple.
 * I am not using TLS or SSL for my LDAP directory. Too much work for this tutorial.

  * The user I created during the installation is: *sysadmin*
  * The password I assigned during the installation is: *12345*

* This local user will be used for all configuration purposes.

Assumptions

* Ubuntu Server 7.10 is installed.
  * No other software was installed during the OS install!
  * After installation you enabled all the repositories in */etc/apt/sources.list*
  * You fully updated your system

```
apt-get update

apt-get upgrade

reboot
```

  * You configured a static IP address. For me I used the following information:

address 192.168.0.60
gateway 192.168.0.1
netmask 255.255.255.0 * You edited your */etc/hosts* file so that your hostname and fully qualified domain name are on the line *127.0.1.1*

```
127.0.1.1 dc01-ubuntu dc01-ubuntu.example.local
```

  * You installed the OpenSSH Server.

```
apt-get install openssh-server
```

  * You did not set a password on the root account. All commands will be run with sudo or by opening a root shell.

```
sudo bash
```

* Currently you do not have any other software running nor do you have any other users on the system.

Step 1: Install WebMin

We will be installing WebMin. Why? I like to use it to configure some things. This step is techinically optional but I feel as though it greatly simplifies administration of the server in the future.

Download the WebMin package from their website.

```
wget http://superb-west.dl.sourceforge.net/sourceforge/webadmin/webmin_1.380_all.deb
```

Install pre-requisite software.

```
apt-get install openssl libauthen-pam-perl libio-pty-perl libmd5-perl libnet-ssleay-perl
```

Install WebMin

```
dpkg -i webmin_1.380_all.deb
```

If the installation is successful you will see a message similar to this:

```
"Webmin install complete. You can now login to https://dc01-ubuntu.example.local:10000/
  as root with your root password,
or as any user who can use sudo to run commands as root."
```

Step 2: Install OpenLDAP

For our LDAP server we will be using the very flexible OpenLDAP Server (slapd).

Install the software.

```
apt-get install slapd ldap-utils migrationtools
```

Answer the on-screen prompts with:

```
Admin password: 12345
Confirm password: 12345
```

We need to configure OpenLDAP now.

```
dpkg-reconfigure slapd
```

Answer the on-screen prompts with:

```
No
  DNS domain name: example.local
  Name of your organization: example.local
  Admin password: 12345
  Confirm password: 12345
  OK
  BDB
  No
  Yes
  No
```

Restart OpenLDAP.

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

Step 3: Install SAMBA

We will be using SAMBA for some main functions in this tutorial. In order to configure OpenLDAP correctly we must first install SAMBA.

Install the software.

```
apt-get install samba smbldap-tools smbclient samba-doc
```

Step 4: Configure OpenLDAP for use with SAMBA

In order to use LDAP and SAMBA we need to configure the */etc/ldap/slapd.conf* file.

Copy the samba.schema file to the OpenLDAP schema directory.

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

Unzip the file.

```
gzip -d /etc/ldap/schema/samba.schema.gz
```

Open the /etc/ldap/slapd.conf file for editing.

```
vim /etc/ldap/slapd.conf
```

Add the following lines to the document where the other "include" lines are:

```
include         /etc/ldap/schema/samba.schema
include         /etc/ldap/schema/misc.schema
```

Change the line:

```
access to attribute=userPassword
```

to:

```
access to attrs=userPassword,sambaNTPassword,sambaLMPassword
```

Restart OpenLDAP:

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

Step 5: Configure SAMBA

Now we need to configure SAMBA. This includes configuring the */etc/samba/smb.conf* file.

Open up the SAMBA directory.

```
cd /etc/samba/
```

Backup the samba configuration file.

```
cp smb.conf smb.conf.original
```

Open the samba configuration file for editing.

```
vim smb.conf
```

Make the following changes throughout the file:

```
workgroup = EXAMPLE
security = user
```

```
passdb backend = ldapsam:ldap://localhost/
obey pam restrictions = no
########################################################################
#COPY AND PASTE THE FOLLOWING UNDERNEATH "OBEY PAM RESTRICTIONS = NO"
########################################################################
#
# Begin: Custom LDAP Entries
#
ldap admin dn = cn=admin,dc=example,dc=local
ldap suffix = dc=example, dc=local
ldap group suffix = ou=Groups
ldap user suffix = ou=Users
ldap machine suffix = ou=Computers
ldap idmap suffix = ou=Users
; Do ldap passwd sync
ldap passwd sync = Yes
passwd program = /usr/sbin/smbldap-passwd %u
passwd chat = *New*password* %n\n *Retype*new*password* %n\n *all*authentication*tokens*updated*
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 -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"
domain logons = yes
#
# End: Custom LDAP Entries
#
########################################################
#STOP COPYING HERE!
```

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

Comment out the line:

```
invalid users = root
```

Add the following line:

```
logon path =
```

Restart SAMBA.

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

Give SAMBA the "admin" password to the LDAP tree.

```
smbpasswd -w 12345
```

Step 6: Configure the SMBLDAP-TOOLS package.

We will be using the smbldap-tools package to populate our directory, add users, add workstations, etc... But, the tools need to be configured first!

Open up the examples directory.

```
cd /usr/share/doc/smbldap-tools/examples/
```

Copy the configuration files to */etc/smbldap-tools*:

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

cp smbldap.conf.gz /etc/smbldap-tools/
```

Unzip the configuration file.

```
gzip -d /etc/smbldap-tools/smbldap.conf.gz
```

Open up the */etc/smbldap-tools* directory.

```
cd /etc/smbldap-tools/
```

Get the SID (Security ID) for your SAMBA domain.

```
net getlocalsid
```

This results in (example): SID for domain DC01-UBUNTU is: S-1-5-21-949328747-3404738746-3052206637

Open the */etc/smbldap-tools/smbldap.conf* file for editing.

```
vim smbldap.conf
```

Edit the file so that the following information is correct (according to your individual setup):

```
SID="S-1-5-21-949328747-3404738746-3052206637" ## This line must have the same SID as when you ran "net getlocalsid"
sambaDomain="EXAMPLE"
```

```
ldapTLS="0"
suffix="dc=example,dc=local"
sambaUnixIdPooldn="sambaDomainName=EXAMPLE,${suffix}"
userSmbHome=
userProfile=
userHomeDrive=
userScript=
mailDomain="example.local"
```

Open the */etc/smbldap-tools/smbldap_bind.conf* file for editing.

```
vim smbldap_bind.conf
```

Edit the file so that the following information is correct (according to your individual setup):

```
slaveDN="cn=admin,dc=example,dc=local"
slavePw="12345"
masterDN="cn=admin,dc=example,dc=local"
masterPw="12345"
```

Set the correct permissions on the above files:

```
chmod 0644 /etc/smbldap-tools/smbldap.conf

chmod 0600 /etc/smbldap-tools/smbldap_bind.conf
```

Step 7: Populate LDAP using smbldap-tools

Now we need to populate our LDAP directory with some necessary SAMBA and Windows entries.

Execute the command to populate the directory.

```
smbldap-populate -u 30000 -g 30000
```

At the password prompt assign your root password:

12345

Verify that the directory has information in it by running the command:

```
ldapsearch -x -b dc=example,dc=local | less
```

Step 8: Add an LDAP user to the system

It is time for us to add an LDAP user. We will use this user account to verify that LDAP authentication is working.

Add the user to LDAP

```
smbldap-useradd -a -m -M ricky -c "Richard M" ricky
```

Here is an explanation of the command switches that we used.

```
-a allows Windows as well as Linux login
  -m makes a home directory, leave this off if you do not need local access
  -M sets up the username part of their email address
-c specifies their full name
```

Set the password the new account.

```
smbldap-passwd ricky
```

```
# Password will be: 12345
```

Step 9: Configure the server to use LDAP authentication.

The basic steps for this section came from the Ubuntu Forums (**http://ubuntuforums.org/showthread.php?t=597056**). Thanks to all who contributed to that thread! Basically we need to tell our server to use LDAP authentication as one of its options. Be careful with this! It can cause your server to break! This is why we always have a backup around.

Install the necessary software for this to work.

```
apt-get install auth-client-config libpam-ldap libnss-ldap
```

Answer the prompts on your screen with the following:

```
Should debconf manage LDAP configuration?: Yes
  LDAP server Uniform Resource Identifier: ldapi://127.0.0.1
  Distinguished name of the search base: dc=example,dc=local
  LDAP version to use: 3
  Make local root Database admin: Yes
  Does the LDAP database require login? No
  LDAP account for root: cn=admin,dc=example,dc=local
LDAP root account password: 12345
```

Open the */etc/ldap.conf* file for editing.

```
vim /etc/ldap.conf
```

Configure the following according to your setup:

```
host 127.0.0.1
base dc=example,dc=local
uri ldap://127.0.0.1/
rootbinddn cn=admin,dc=example,dc=local
bind_policy soft
```

Copy the */etc/ldap.conf* file to */etc/ldap/ldap.conf*

```
cp /etc/ldap.conf /etc/ldap/ldap.conf
```

Create a new file */etc/auth-client-config/profile.d/open_ldap*:

```
vim /etc/auth-client-config/profile.d/open_ldap
```

Insert the following into that new file:

```
[open_ldap]
nss_passwd=passwd: compat ldap
nss_group=group: compat ldap
nss_shadow=shadow: compat ldap
pam_auth=auth       required     pam_env.so
 auth       sufficient   pam_unix.so likeauth nullok
 auth       sufficient   pam_ldap.so use_first_pass
 auth       required     pam_deny.so
pam_account=account    sufficient   pam_unix.so
 account    sufficient   pam_ldap.so
 account    required     pam_deny.so
pam_password=password   sufficient   pam_unix.so nullok md5 shadow use_authtok
 password   sufficient   pam_ldap.so use_first_pass
 password   required     pam_deny.so
```

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

Backup the */etc/nsswitch.conf* file:

```
cp /etc/nsswitch.conf /etc/nsswitch.conf.original
```

Backup the */etc/pam.d/* files:

```
cd /etc/pam.d/

mkdir bkup

cp * bkup/
```

Enable the new LDAP Authentication Profile by executing the following command:

```
auth-client-config -a -p open_ldap
```

Reboot the server and test to ensure that you can still log in using SSH and LDAP.

```
reboot
```

Step 10: Install BIND (DNS Server)

Because we are going to be a domain controller and source for authentication it makes sense to also have some DNS services available. Please note that if

you have multiple servers at your disposal it is recommended to install a seperate DNS server as well so we have two to look at.

Install the software.

```
apt-get install bind9
```

Step 11: Configure our primary DNS Zone using WebMin

We now want to create our DNS zone so that we are in charge of it and can make use of it. I prefer using a GUI to do this as opposed to editing the zone files.

In a web browser navigate to: *https://192.168.0.60:10000* (Please use the IP address that YOU assigned to your server.)
Login as "sysadmin" and "12345".
Servers > BIND DNS Server
Under "Existing DNS Zones" click "Create master zone".

```
Zone type: Forward (Names to Addresses)
  Domain name / Network: example.local
  Records file: Automatic
  Master server: dc01-ubuntu.example.local
Email address: sysadmin@example.local
```

Click "Create" button.

Click "Apply Changes" button.

Click "Address (0)" at the top.

```
Name: dc01-ubuntu
  Address: 192.168.0.60
  Click "Create" button
Click "Return to record types"
```


Click "Apply Changes" button.

Step 12: Configure the server to use itself for DNS

DNS doesn't do a whole lot of good if we don't use it. In this section we point our */etc/resolv.conf* file to ourselves. I also recommend leaving in a known working DNS server as the seconday source just in case something screws up. In some of my trials I did notice that the server would hang trying to start BIND9.

Open the */etc/resolv.conf* file for editing.

```
vim /etc/resolv.conf
```

Add the following lines to the beginning of the file:

```
search example.local
nameserver 192.168.0.60
```

Reboot the server to ensure that DNS is working correctly.

```
reboot
```

Step 13: Add a workstation account to LDAP

This tutorial is meant to create an opensource domain for Windows XP Professional client (and Linux clients) to authenticate against. Therefore we will add a workstation account for the Windows XP Professional workstation that we will be joining to the domain.

Execute the command:

```
smbldap-useradd -w client-winxp
```

*** "client-winxp" is the hostname of the computer that you will be adding to the domain. This must be very specific!**

Step 14: Configure your Windows XP Professional Client

Now I will walk you through configuring your Windows XP Professional workstation so that it will join the domain.

Assumptions:

* This is a vanilla installation of Windows XP Professional SP2.
  * The computer name was set during installation to be: *client-winxp*
  * The Administrator password assigned is: *12345*
  * All other installation options have been left at their default settings.
  * After the installation the following occurred:
  * The only user account on the computer in use was "Administrator"
  * All available Windows Updates were installed.
* A static IP address was assigned with the following information (for my setup only!)

```
IP Address: 192.168.0.61
  Gateway: 192.168.0.1
  Netmask: 255.255.255.0
  DNS: 192.168.0.60
  Search domain: example.local
```

Join the workstation to the domain.

* Log into the computer as Administrator.
  * Right click "My Computer" and click "Properties".
  * Click the tab "Computer Name".
  * Click the button labeled "Change".
  * At the bottom click the radial button labeled "Domain".
  * In the box type the word "example" without quotes!
  * Click the "OK" button.
  * At the password prompt enter "root" for the user and "12345" for the password (substitute the password for what you assigned to your root user earlier!).

It should say "Welcome to the example domain."
* Click "OK".
* Click "OK" again.
* Click "OK" again.
Restart the workstation.

Log in with your test user ("ricky") from earlier.
Try logging into the Windows XP workstation (after selecting the domain from the drop down box) using our test user. It should work without issue!

Notes
Please note that this is basic authentication right now. You're on your own if you wish to add logon scripts, mapped drives, etc...

Step 15: (Optional) Install Apache2 and PHPLDAPAdmin

A nice way to view and modify your LDAP tree is with a GUI. PHPLDAPAdmin is one that many people recommend so I will show you how to install it and use it.

Install the software.

```
apt-get install apache2 phpldapadmin
```

Open the file */etc/apache2/httpd.conf* for editing:

```
vim /etc/apache2/httpd.conf
```

Add the following line to the top of the file. This prevents an annoying error message from Apache2.

```
ServerName dc01-ubuntu.example.local
```

Restart Apache2

```
/etc/init.d/apache2 restart
```

Copy the PHPLDAPAdmin folder into the main web site directory. This is the lazy way of doing things. This way we don't need to create a virtual server, we just access PHPLDAPAdmin by going to: *http://192.168.0.60/phpldapadmin/*

```
cp -R /usr/share/phpldapadmin/ /var/www/phpldapadmin
```

There you have it! A full Ubuntu LDAP and SAMBA Domain Controller in 15 easy steps.