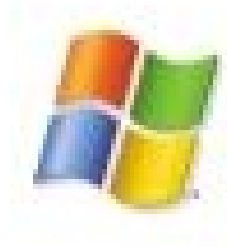


K_{12LTSP}
Authentication
Against
Windows **A**ctive **D**irectory

From K12LTSP/Fedora Core Installation to Windows ADS Authentication



Paul VanGundy
pvangundy@gmail.com
Copyright © 2006 Paul VanGundy
Version 1.2 – April 2006

Setting up and installing K12LTSP is fairly easy. The difficulty comes into play when you want to get your K12LTSP server to authenticate against a Windows 2000/2003 Active Directory server.

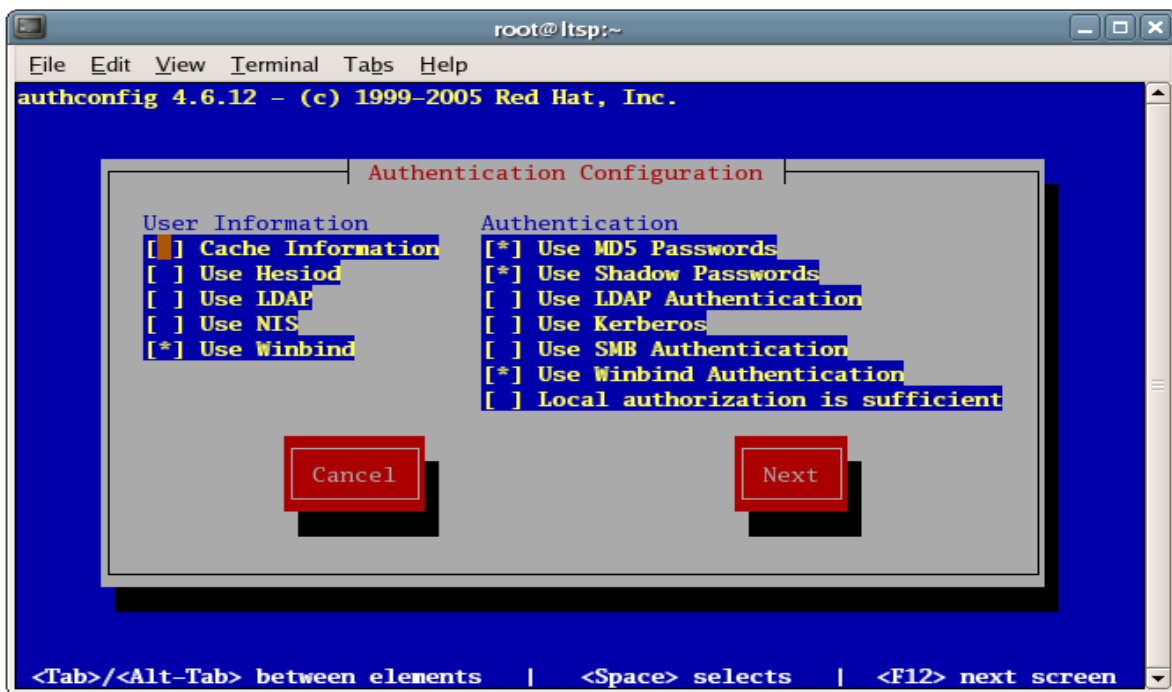
When I originally thought about installing K12LTSP I wanted to make sure that it wasn't going to be more difficult to operate than a Windows box for our users. Part of getting users to try and except Linux is making it as painless as possible and making it work seamlessly with Windows computers. That's where this guide will come in handy as I will show you what steps you need to take and what configuration files you need to modify in order for this to work. The version of K12LTSP I am basing this tutorial off of is 4.4.1.

NOTE: During the installation of K12LTSP ensure that you have named your server with a FQDN. EX: If my box is named LTSP and I am joining to a domain named computers.local then I need to ensure that my computer name is ltsp.computers.local and not just LTSP. This is very important.

Step 1: authconfig

Part I

The majority of modifications is done directly in the authconfig configuration utility. To access authconfig open a terminal window and type 'authconfig' without the ''. This will open authconfig right inside of your terminal window. This is where the fun begins. Look at my authconfig below. This is how your authconfig should like.

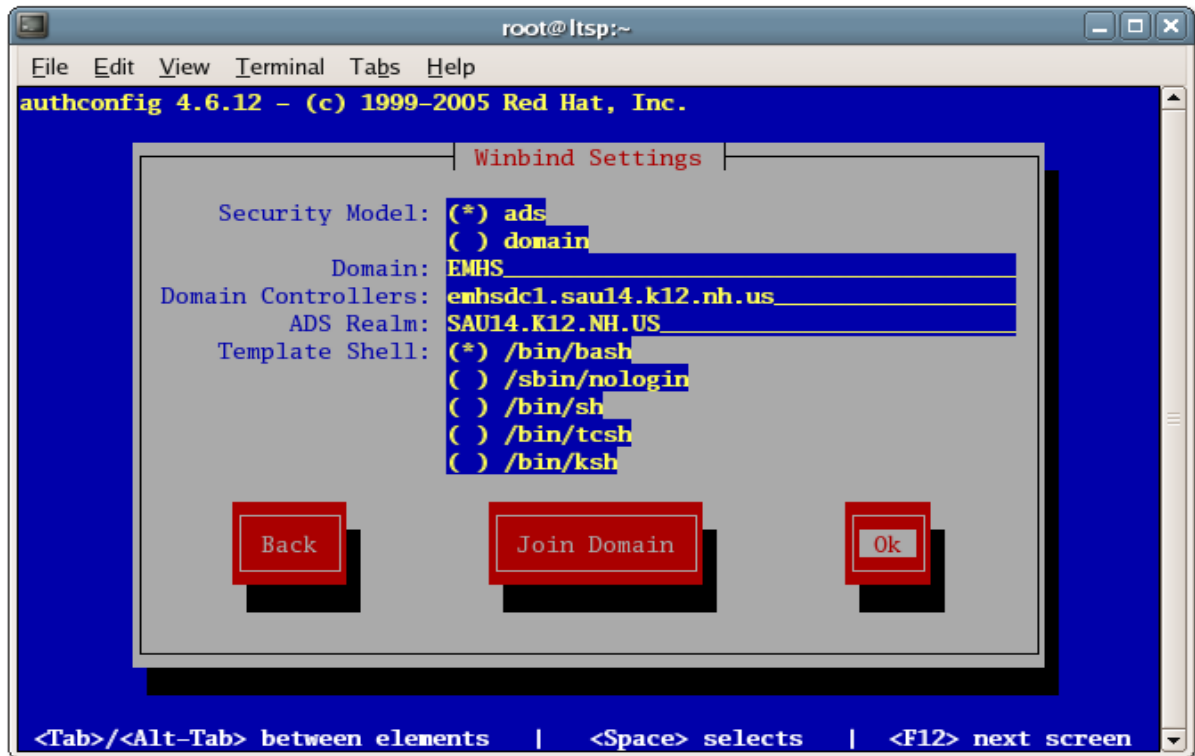


Let me explain what you are viewing. As you can see under 'User Information' we are going to use Winbind to log in our users.

Under 'Authentication' we will leave MD5 Passwords and Shadow Passwords checked. Let me explain why. When 'Use Shadow Passwords' is checked the passwords are stored as hashes in the /etc/shadow file instead of

the /etc/passwd file. This allows for passwords to be more secure. We leave 'Use MD5 Passwords' checked because MD5 uses a hashing algorithm. This allows for passwords to be more cryptic therefore more secure. Finally, we will check the 'Use Winbind Authentication'. The reason why we select this option is because we are telling our server to authenticate via Winbind versus against local files. After you have selected the proper 'User Information' and 'Authentication' methods tab to the next bottom to continue. You will then see the screen below:

Part II



This is the second part of authconfig. Above you see that I have selected my security model to ads. The ads model means you are going to authenticate against a Windows Active Directory Server (ADS). The domain option would mean you are authenticating against a PDC/BDC server. So after you select ads you will then enter your domain name. This should be the short name for your domain. As you see above my is EMHS. In the domain controllers you should put your the fully qualified domain name (FQDN) of the domain controller here. As you can see, mine is emhsdc1.sau14.k12.nh.us. In the ADS Realm you will put your FQDN in all caps like above. Finally, you will select the template shell to be /bin/bash. /bin/bash is the default Unix shell. When you are done, tab to OK and hit Enter. IMPORTANT: DO NOT JOIN DOMAIN. We still have other files we must modify in order to get this to work.

Step 2: Configuring the configuration files

Part I: Automatic Creation of user Home Directories

Every Unix user needs a Home Directory. Home Directories store user information in them. Without a home directory there is no where to store user information. There are several files that we are going to edit in order to get this to work correctly. The files are:

- /etc/pam.d/gdm (If KDE is your default desktop environment then edit the kdm file.)
- /etc/pam.d/login
- /etc/pam.d/sshd

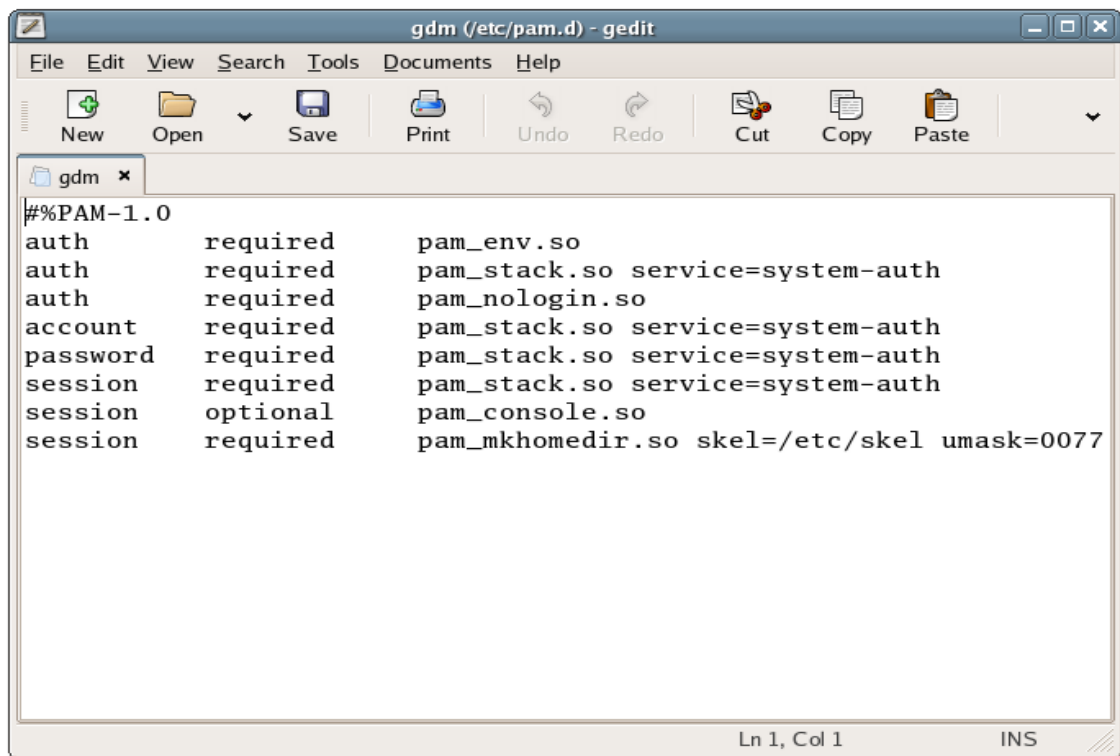
As you can see they are all located in the /etc/pam.d/ directory. This directory holds configuration files that you can modify to configure PAM.

In each file listed above you will want to add the following line:

```
session    required    pam_mkhome.so skel=/etc/skel umask=0077
```

This will allow for automatic creation of user directories when a user logs in. So let's take a look at how each file should look.

/ETC/PAM.D/GDM



```
gdm (/etc/pam.d) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste
gdm x
#%PAM-1.0
auth      required    pam_env.so
auth      required    pam_stack.so service=system-auth
auth      required    pam_nologin.so
account   required    pam_stack.so service=system-auth
password  required    pam_stack.so service=system-auth
session   required    pam_stack.so service=system-auth
session   optional    pam_console.so
session   required    pam_mkhome.so skel=/etc/skel umask=0077
Ln 1, Col 1  INS
```

As you can see above we added the pam_mkhome.so line at the end of the GDM file. This now tells the system to automatically create home directories when a user logs in via GNOME Display Manager (GUI).

/ETC/PAM.D/LOGIN

```
login (/etc/pam.d) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste
login x
#%%PAM-1.0
auth required pam_securetty.so
auth required pam_stack.so service=system-auth
auth required pam_nologin.so
account required pam_stack.so service=system-auth
password required pam_stack.so service=system-auth
# pam_selinux.so close should be the first session rule
session required pam_selinux.so close
session required pam_stack.so service=system-auth
session required pam_loginuid.so
session optional pam_console.so
session required pam_mkhomedir.so skel=/etc/skel umask=0077
# pam_selinux.so open should be the last session rule
session required pam_selinux.so multiple open
Ln 1, Col 1 INS
```

As you can see above, we added the line `pam_mkhomedir.so` right before the part that says “`# pam_selinux.so open should be the last session rule.`” Since we added `pam_mkhomedir.so` to this configuration that tells the system to automatically create home directories when someone logs in via text based virtual consoles.

/ETC/PAM.D/SSHD

```
sshd (/etc/pam.d) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste
sshd x
#%%PAM-1.0
auth required pam_stack.so service=system-auth
account required pam_nologin.so
account required pam_stack.so service=system-auth
password required pam_stack.so service=system-auth
session required pam_stack.so service=system-auth
session required pam_loginuid.so
session required pam_mkhomedir.so skel=/etc/skel umask=0077
Ln 1, Col 1 INS
```

Again, we added the pam_mkhomedir.so line at the end of our sshd configuration file. This tells the system to automatically create home directories when a user logs in through secure means remotely with SSH.

What is skel and umask?

skel – setting the skel to a certain directory ensures that those files listed in that directory are automatically copied into the home directory of a new user when they are created. Since /etc/skel is an empty directory by default it ensures our new users will have an empty home directory.

umask – determines the permissions that are assigned to a new home directory. However, the permissions can only be set by octal format (1, 2, 4). 1 represents executable, 2 represents writable, and 4 represents readable. You can use them as they are or combine them (EX: 1+2+4= 7 which gives the permission to have executable, writable and readable permissions. There are four columns represented in the umask. The farthest right column represents user/owner permissions. Second from the right represents group permissions. Third from the right represents permissions for everyone. The fourth from the right is always a 0. So in our pam_mkhomedir.so we added to the gdm, login and sshd files umask=0077 means we gave execute, write, and read privileges to both owner and group.

Step 3: Almost done

We're down to the final steps. The last things that we must do is configure Samba to know where home directories are to be located and then join the AD!

Part I:

Now what we want to do is create the location where our user home directories will be created. This is needed because Samba tells the system where to create home folders when a user logs in via AD authentication. Open up a terminal and type the following line:

```
mkdir /home/*DOMAINNAME*
```

Substitute DOMAINNAME with the name of your domain. For example, my domain is EMHS so I would type the following

```
mkdir /home/EMHS
```

Now we need to set the proper permissions on our new folder. Inside of the terminal type the following line

```
chmod a-rwx,a+rx,u+rwx /home/*DOMAINNAME*
```

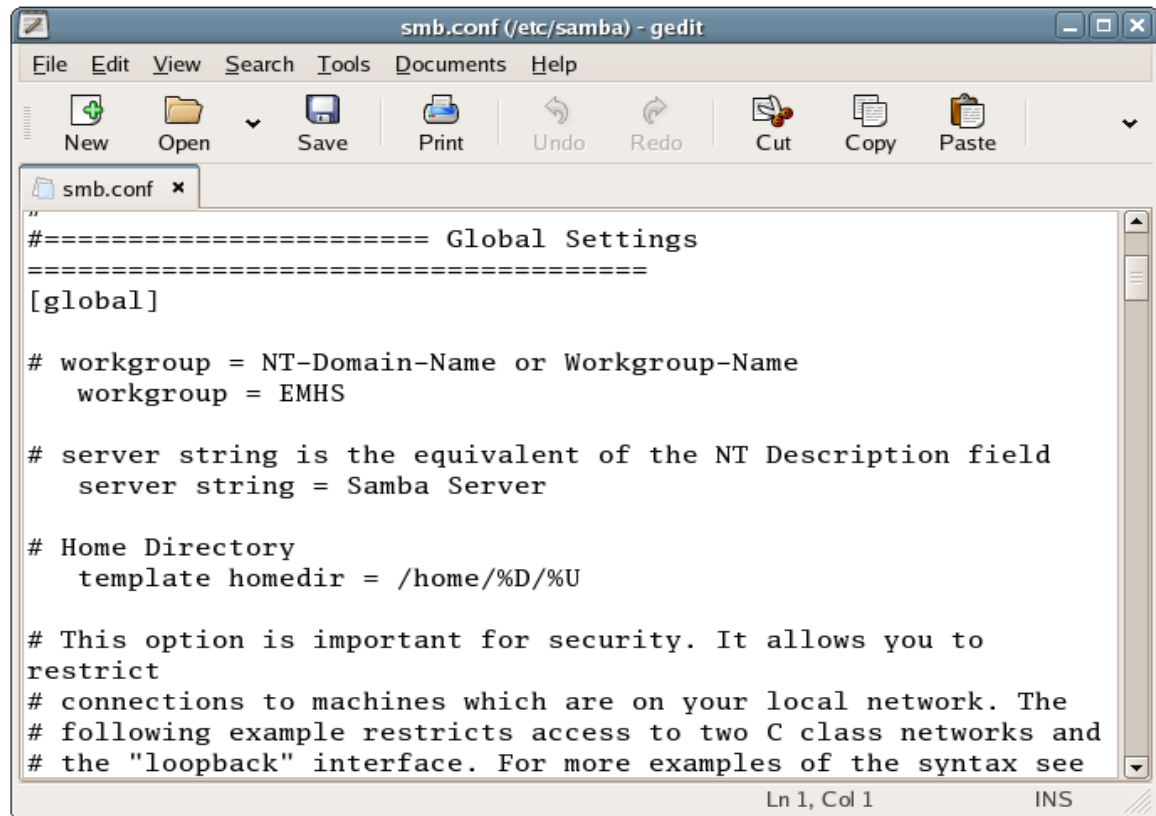
The above just set permissions on our newly created domain folder so that everyone can read and execute the contents but only root can read, execute and write to the contents.

Part II:

With the text editor of your choice open `/etc/samba/smb.conf`. Find the area where it says `[global]`. All we want to do in this file is add the following line into that area:

template homedir = /home/%D/%U

To see an example view the picture of my `smb.conf` file below:



```
##### Global Settings
#####
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
workgroup = EMHS

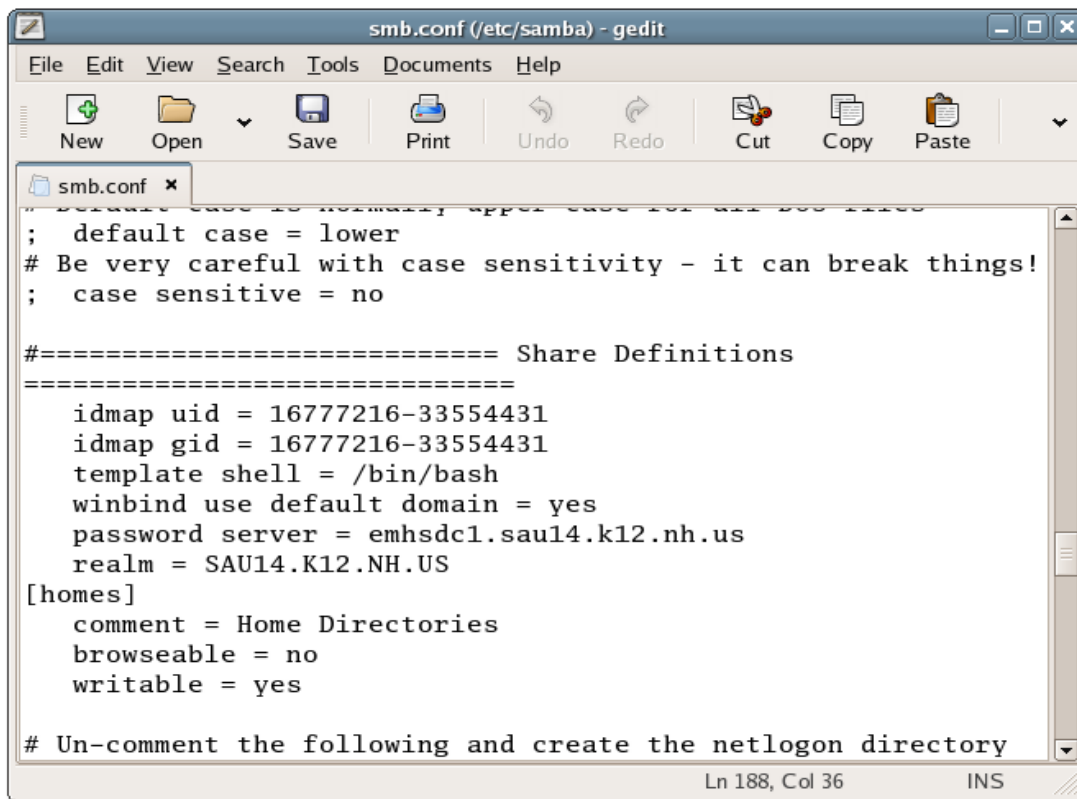
# server string is the equivalent of the NT Description field
server string = Samba Server

# Home Directory
template homedir = /home/%D/%U

# This option is important for security. It allows you to
restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
```

As you can see above I added a comment line after the server string line that says “# Home Directory” so that way I could easily find where I put the `template homedir = /home/%D/%U`. So essentially what this line is saying is create a home directory in the `/home/*DOMAIN*/`. Using my domain as an example it is saying `/home/EMHS`. So when a new user named `sjohn` logs in for the first time it will create the `/home/EMHS/sjohn` home directory.

The last thing that you are going to want to change in the smb.conf file is whether Winbind is to use the default domain listed or not. You do this by going to the section called Share Definitions. The option we are looking for is about the 4th line down and right below template shell = /bin/bash. By default it is set to no and we want to change it to yes. Look below to see how I set Winbind to use our default domain:



```
; default case = lower
# Be very careful with case sensitivity - it can break things!
; case sensitive = no

===== Share Definitions =====
; idmap uid = 16777216-33554431
; idmap gid = 16777216-33554431
; template shell = /bin/bash
; winbind use default domain = yes
; password server = emhxdc1.sau14.k12.nh.us
; realm = SAU14.K12.NH.US
[homes]
; comment = Home Directories
; browseable = no
; writable = yes

# Un-comment the following and create the netlogon directory
```

Step 4: Joining the ADS

Part I:

We finally made it. It's time to join our Active Directory domain. Here are the steps we need to do:

1. Open terminal and type the following: **service winbind restart**
This restarts the Winbind service with our new changes
2. Inside the terminal type the following: **net ads join -U administrator**
You will be prompted for the administrator password. Put in the domain administrator's password

That's it. You have joined your K12LTSP server to your Windows Active Directory server. To test if it worked you can do the following:

- Open a terminal and type the following commands:
wbinfo -u – This should return a list of all users in your Windows ADS.
wbinfo -g – This should return a list of all the groups in your Windows ADS.
- You can always try to log in as a domain user to check.

If it didn't work check the following things:

- Is the time on your K12LTSP box the same as the time on your domain controller?
- Is Active Directory allowing Windows boxes to authenticate?
- Are you sure you gave your computer a FQDN? Example was ltsp.computers.local instead of just LTSP.
- If there is a failed attempt view the `/var/log/messages` file.

I hope this tutorial has been helpful in assisting you with joining your K12LTSP server to your present Active Directory environment. Now you can have Linux boxes and Windows boxes side by side authenticating to the same server.