Network-based Identity Management and LDAP

License

Contents

General notes about the labs

Often the lab instructions are intentionally open ended, and you will have to figure some things out for yourselves. This module is designed to be challenging, as well as fun!

However, we aim to provide a well planned and fluent experience. If you notice any mistakes in the lab instructions or you feel some important information is missing, please feel free to add a comment to the document by highlighting the text and click the comment icon (), and I (Cliffe) will try to address any issues. Note that your comments are public.

The labs are written to be informative and, in order to aid clarity, instructions that you should actually execute are generally written in this colour. Note that all lab content is assessable for the module, but the colour coding may help you skip to the "next thing to *do*", but make sure you dedicate time to read and understand everything. Coloured instructions in *italics* indicates you need to change the instructions based on your environment: for example, using your own IP address.

You should maintain a **lab logbook / document**, which should include your answers to the questions posed throughout the labs (in this colour).

Preparation

As with all of the labs in this module, start by loading the latest version of the LinuxZ template from the IMS system. If you have access to this lab sheet, you can read ahead while you wait for the image to load.

> To load the image: press F12 during startup (on the boot screen) to access the IMS system, then login to IMS using your university password. Load the template image: LinuxZ (load the latest version).

Once your LinuxZ image has loaded, log in using the username and password allocated to you by your tutor.

The root password -- **which should NOT be used to log in graphically** -- is "tiaspbiqe2r" (**t**his **i**s **a** **s**ecure **p**assword **b**ut **i**s **q**uite **e**asy **2** **r**emember). Again, never log in to the desktop environment using the root account -- that is bad practice, and should always be avoided.

Using the VM download script (as described in a previous lab), download and **start** these VMs:

- WinXP Pro SP3 with pGina - bridged with network share
- LinuxY 32bit openSUSE 13.1 KDE 4 desktop
  username: student, password: theIliad
  username: root, password: tiaspbiqe2r
- SUSE Linux Enterprise Server SLES 11 SP3 - server
  username: root, password: toor
- Kali Linux - Live Disk (alternatively, you can use LinuxZ for this part of the task)
  username: root, password: toor

Configure each VM so that it is on a shared network (for example, all set to bridged).

Introduction to managing identity and authentication across a network of computers

It is not unusual for an organisation's network to include a mix of *many* Unix and/or Windows computers. An organisation also often maintains a database of users or employees, with details such as email addresses and phone numbers. Many organisations prefer to have a system where users have user accounts centrally managed, so that users can log in to various systems using the same login details.

Due to the long history of technical solutions to these goals, there are many ways these can be achieved. Each solution has security and usability advantages and disadvantages.

In this lab you will implement centralised network-based authentication using LDAP, and you will investigate the security consequences. To save time, you will be making use of automated/graphical configuration tools, although if you end up specialising in this in the future, you will want to familiarise yourself further with the command line tools and relevant configuration files.

Creating your own Certificate Authority

In order to use encrypted communications to secure our server, you need to set up a certificate for the server.

A certificate authority (CA) is a trusted third party that issues digital certificates. A digital certificate, such as an **X.509 certificate** (as used with most encrypted Internet communications, such as encrypted websites or FTP), specifies details about a server and its *public key*. Public key infrastructure (PKI) schemes use *public keys*, shared with the world, and *private keys* that are kept secret.

If the certificate has been *signed* by a CA that the client trusts, then the server can prove its identity to the client. Without a third party (the CA) to verify the identity, it is hard to know whether a man-in-the-middle (MITM) attack is taking place.

Typically a public server will pay a well known CA to sign their certificate. However, for certificates that are only used within an organisation, a new CA can be created and deployed to the computers to establish trust in the certificates signed by the new CA.

On the SUSE Linux Enterprise Server (SLES) VM:

Accept any licences, by pressing "q", then answering "y"es.

Login as username "root", password "toor".

Make a note of the server's IP address ("ifconfig", should start with 172.16.).

Start YaST (Yet another Setup Tool), which is SUSE's configuration tool:

```
yast2
```

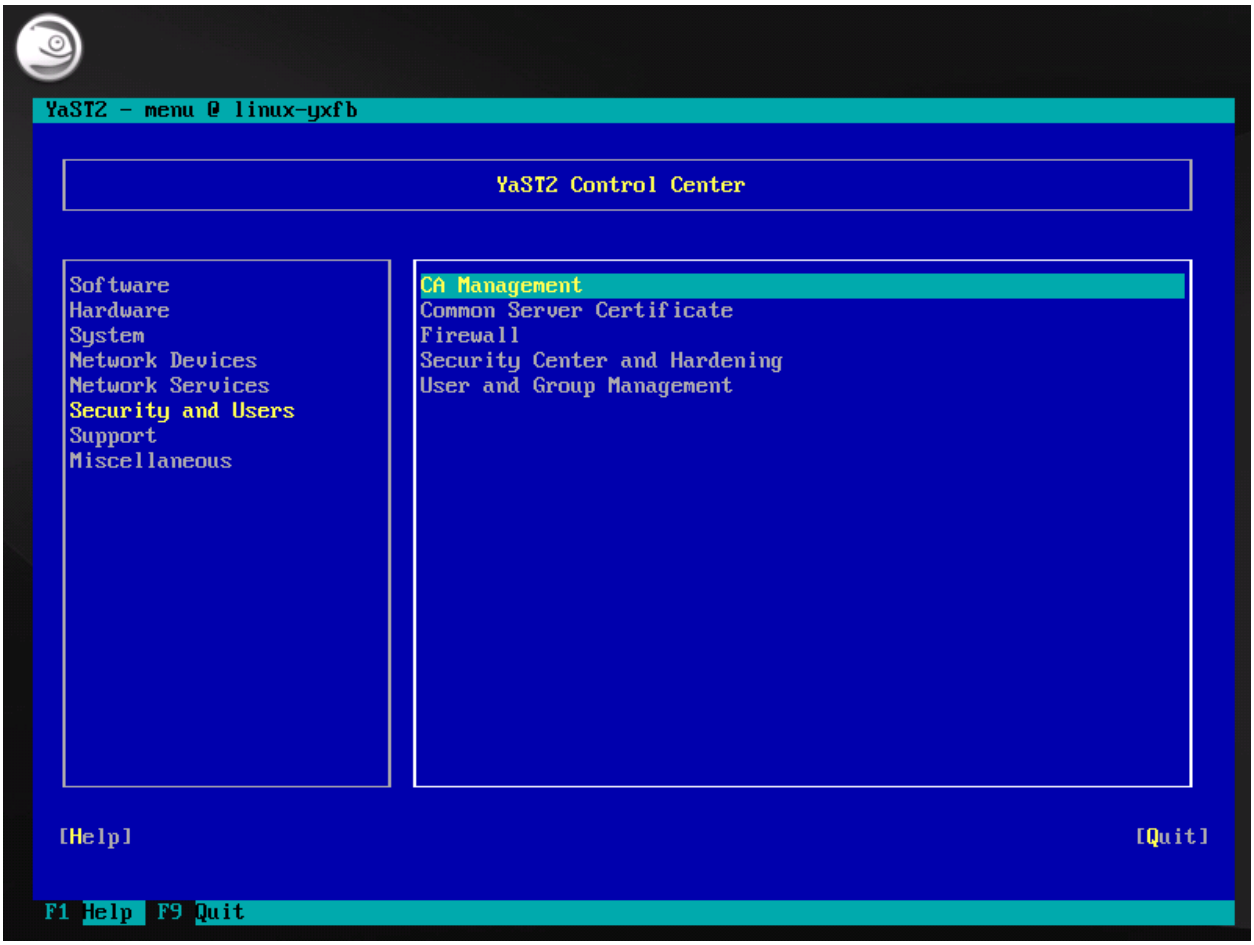
Note that one of the nice features about YaST, is that on systems with a graphical desktop, YaST displays as a graphical program, and on a server without X, it displays the same features using ncurses (the console-based graphical interface as shown below).

Navigation tips: use Tab to move to next interface item, Space to check boxes, Enter to select/press buttons. Shortcuts: use Alt + the highlighted letter, for example, "Alt+N" for "**N**ext".

Creating CAs and certificates

You will now create a your own CA…

Navigate to "Security and Users" / "CA Management".



YaST with ncursors interface, starting CA Management

Select “Create Root CA”.

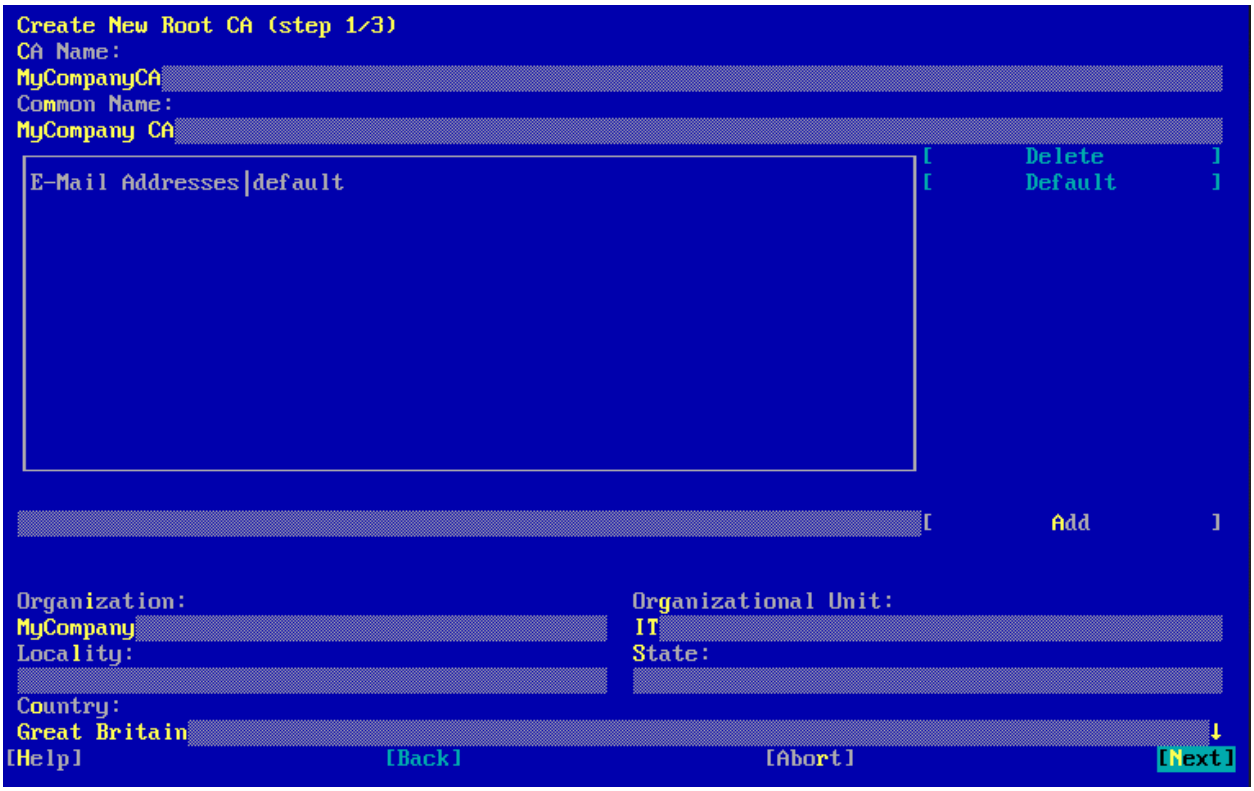
Under CA Name enter a short name for the new CA, such as “MyCompanyCA”.

Under Common Name enter a name (this can include spaces), such as “MyCompany CA”.

Under Organisation enter a company name, such as “MyCompany”.

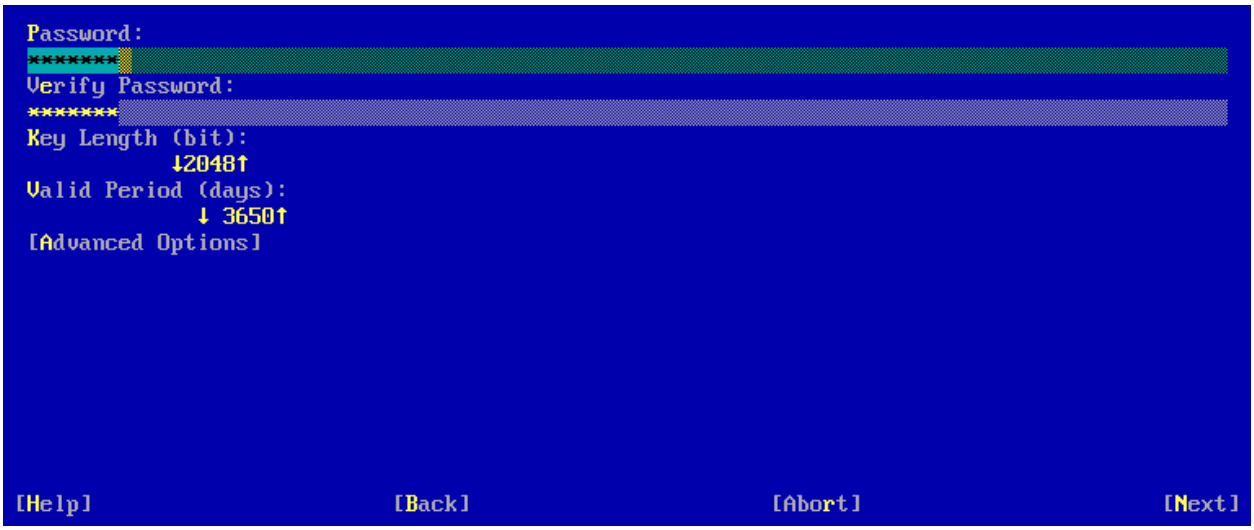
Under Organisational Unit enter a department name, such as “IT”.

Select “Next”.



YaST CA Management, creating a new CA

Enter a password (make a note of the password you choose).



YaST CA Management, entering a password for a new CA

If you like you can look through the Advanced Options, and optionally change the nsComment field.

Select “Next”.

Review the details and select “Create”.

YaST CA Management, entering a CA

Next, “Enter CA” to use this CA to create and sign a new certificate.

Enter the password you noted earlier, when creating the CA.

Navigate to the Certificates tab.

Select “Add”.

Select “Add Server Certificate”.

For Common Name enter the name of the computer (shown at the top of the YaST window). For example, “linux-yxfb”.

YaST CA Management, creating a server certificate

Select “Next”.

Enter a password (make a note of the password you choose).

Select “Next”.

Review the details and select “Create”.

Select “Export”.

Select “Export as Common Server Certificate”.

Tip: an error message about the server name not matching the certificate can either be ignored, or to fix: edit /etc/hosts and edit the line “127.0.0.1 localhost” to “127.0.0.1 ***linux-yxfb*** localhost”, where *linux-yxfb* is the server’s hostname.

Select “OK”.

If you were to use this certificate on the Internet, would current Web browsers, such as Firefox, trust the connection? Why not?

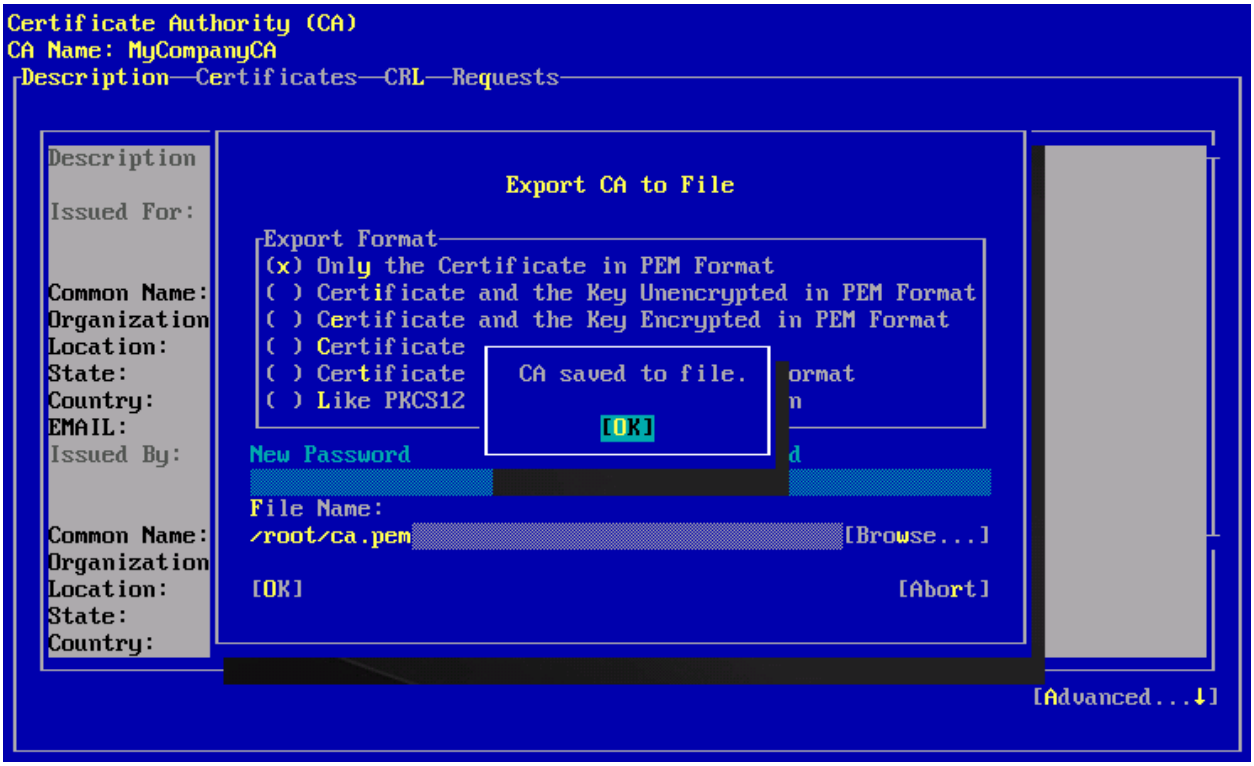
How would you make Firefox trust this certificate?

How would you get a certificate that Firefox would trust as-is?

Next, “Enter CA” and export the CA certificate, so you can share this with your clients.

Under “Advanced”, select “Export CA to File”.

Check “Only the Certificate in PEM Format” option, and for File Name enter “/root/ca.pem”.



YaST CA Management, exporting the public CA certificate

Select “OK”.

Select “Finish”.

On the LinuxY 32bit openSUSE 13.1 KDE 4 desktop VM (or on LinuxZ):

Login as a normal user (the “student” account). Note, you may be automatically logged in.

In a console, ensure the graphical version of YaST is installed (may require a bridged network interface):

```
sudo zypper install yast2-qt
```

Copy the server certificate to this host via SSH:

```
sudo mkdir /usr/local/share/ca-certificates
```

```
sudo scp root@SERVER-IP:/root/ca.pem /usr/local/share/ca-certificates/
```

(Where SERVER-IP is the IP address of the server VM)

```
sudo /usr/sbin/update-ca-certificates
```

Introducing Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is a network protocol for accessing a hierarchical (tree-structured) database of information over TCP/IP. Technically, LDAP is not the term for the database, but the *protocol* that is used to talk to one; in practice LDAP server software tend to implement a database. Typically an organisation would use LDAP to share "phonebook" information about employees, such as their email addresses, job titles, office numbers, and so on. LDAP can also be used to centrally store credentials, so that people can use the same username and password on any computer that authenticates against the LDAP server. Other common uses includes configuration of DNS, DHCP, and email servers.

By default LDAP uses TCP port and UDP port 389. LDAP commands include searching, adding, modifying or deleting entries in the database. LDAP is a binary protocol (meaning it does not send ASCII human-readable instructions), and is by default not encrypted (information is plain text), which means that an eavesdropper (using a sniffer such as Wireshark) can view any information sent via LDAP.

LDAP has since been updated to include encryption add-ons, such as Transport Layer Security (TLS)/SSL, and can also be tunnelled through SSH. Note that some LDAP clients do not check the server's domain, but only provide encryption (meaning MITM attacks may be possible).

LDAP has some access control features, so that a client may be restricted to certain operations or data.

Setting up your own LDAP server

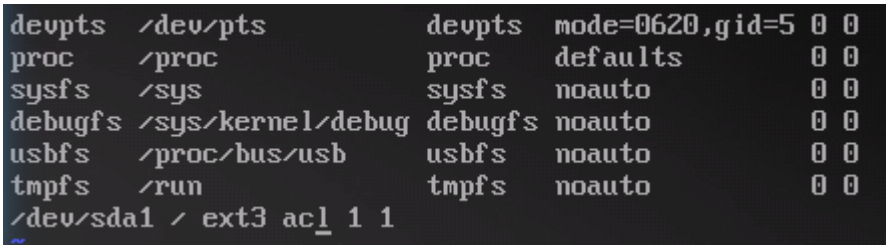
On the SUSE Linux Enterprise Server (SLES) VM:

LDAP on SUSE requires ACL support, so enable it:

```
vi /etc/fstab
```

Edit the options used for mounting "/" when the system boots to include "acl", as shown below (replace "defaults" with "acl").

Reminder: Vi is 'modal': it has an insert mode, where you can type text into the file, and normal mode, where what you type is interpreted as commands. Press the "i" key to enter "insert mode". Type your changes to the file, then exit back to "normal mode" by pressing the Esc key. Now to exit and save the file press the ":" key, followed by "wq" (write quit), and press Enter.



Editing /etc/fstab, to enable ACL support

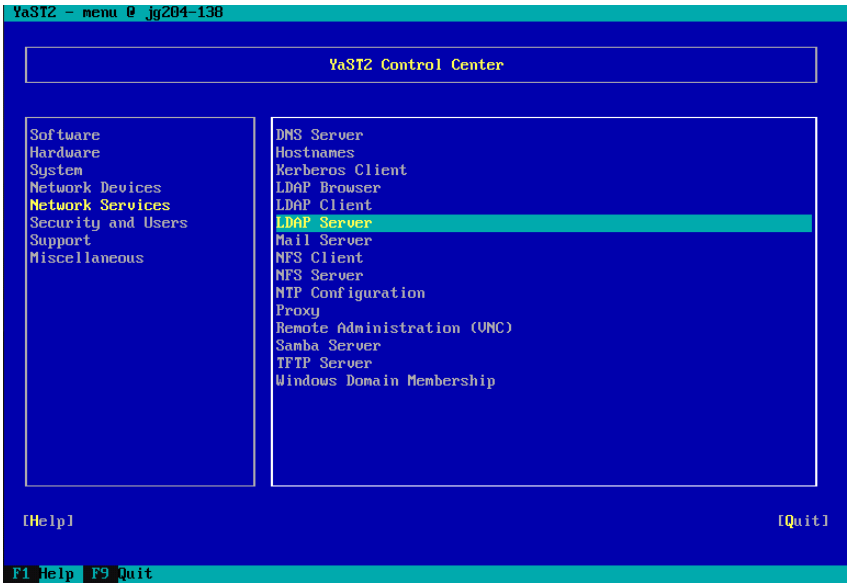
Remount "/" with acl support now:

```
mount -o remount,acl /
```

Start YaST:

```
yast2
```

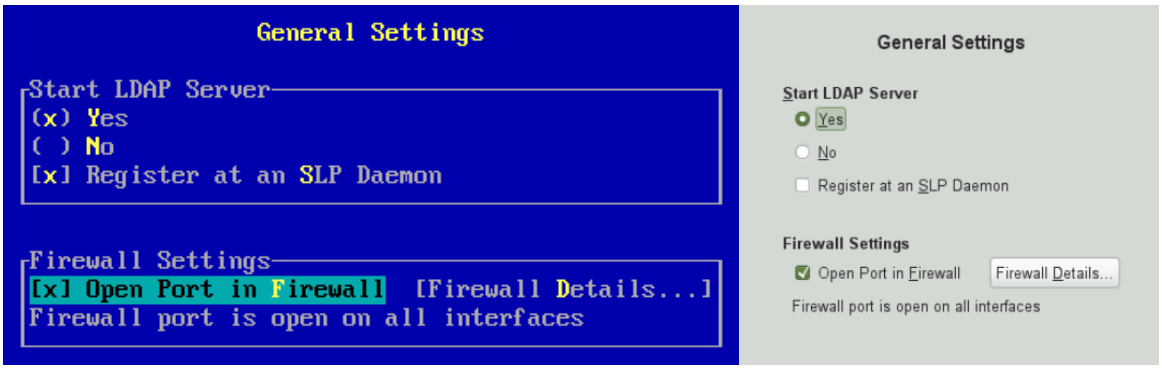
Navigate to "Network Services", "LDAP Server".



YaST LDAP Server Module

Choose to start the server ("Yes") and check "Register at an SLP Daemon" (which can be used by clients to find your new LDAP service).

Open the port in the firewall (by checking this option).



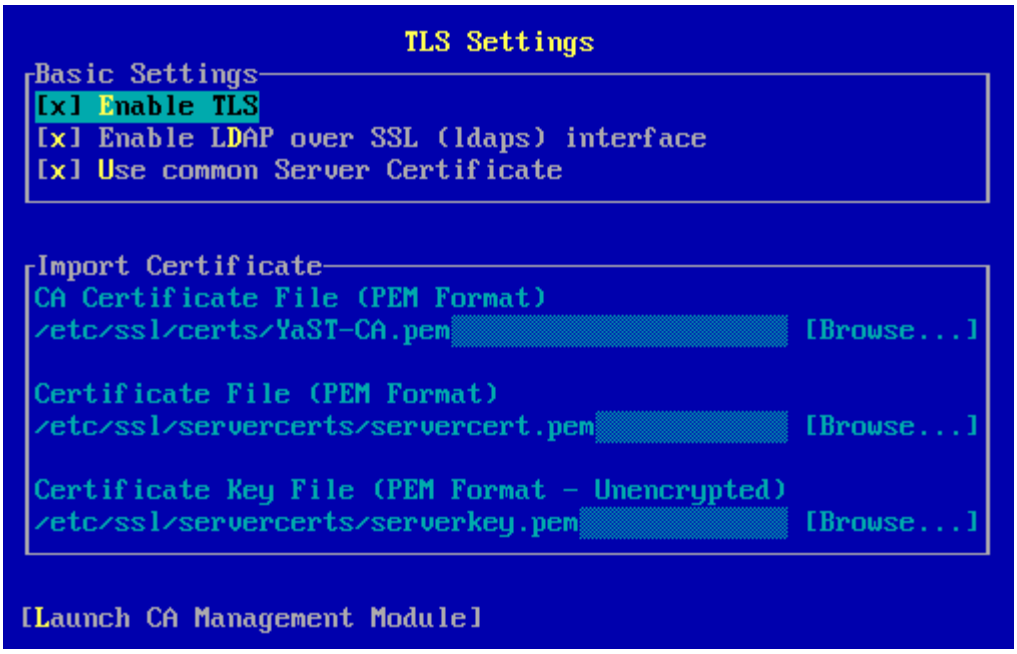
Yast LDAP Server Config, ncusors and Qt graphical interfaces side-by-side

Select "Next".

Choose "Stand-alone server". If you were managing a large network you could configure replication between multiple servers.

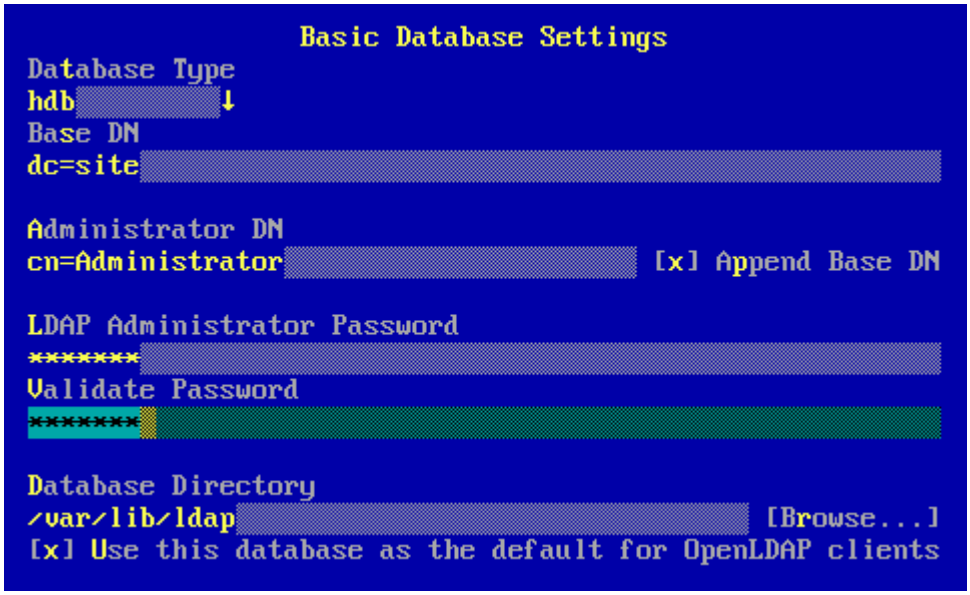
Enable all TLS Settings, so encryption options are selected. It should automatically detect the server certificate you created earlier.

Select "Next".



YaST LDAP Server encryption settings

Leave the basic database settings as they are (as shown below), and enter an LDAP administration password, make a note of the password you choose.



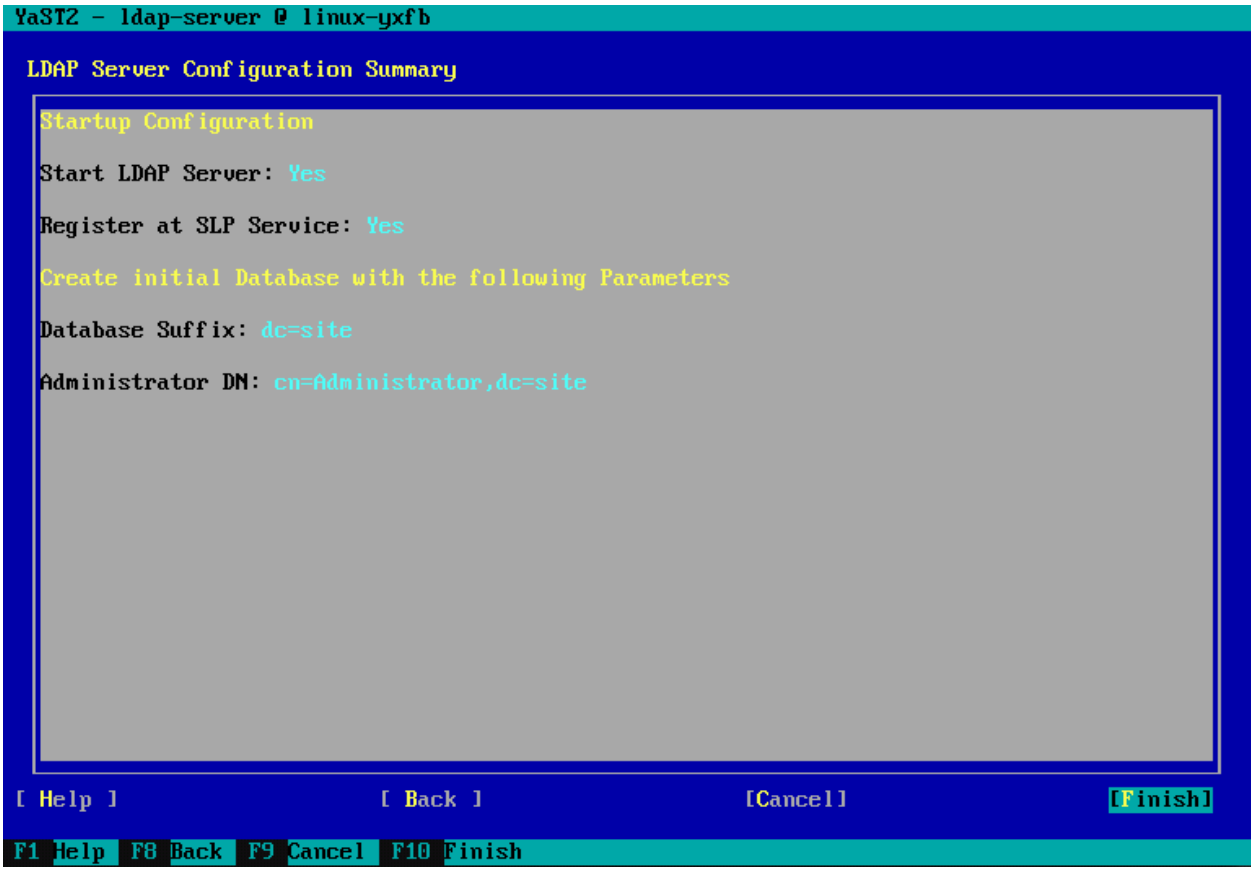
YaST LDAP Server database settings

Select “Next”.

A summary of your settings will be displayed.

Select “Finish”.

> If you get an error message regarding saving the configuration, try redoing the above a second time.



YaST LDAP Server configuration complete

At this point you have a minimal LDAP server running, although the database is currently empty.

If you reopen the YaST LDAP Server module, you can view further configuration and management options.

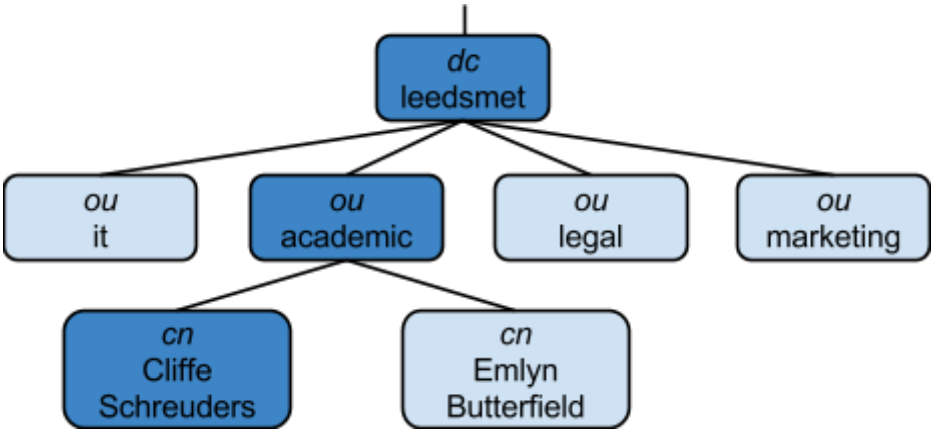
Storing and retrieving in LDAP

The hierarchy (tree) of information stored via LDAP is known as the *directory information tree* (DIT). The structure of the database is defined via a *schema*. For example, a schema may define the fact that the database stores information about people, and what kinds of data is valid.

The root (main) node is commonly a set of "**dc**" values, which is short for *domainComponent*. For example, for leedsmet.ac.uk, "dc=leedsmet, dc=ac, dc=uk". In the database you created earlier, the dc="site".

Other common values are the "**ou**" or *organizationalUnit*, which defines the department within an organisation, such as "it" or "academic", and also the "**cn**" or "**sn**" (common name / surname) is used to describe the way an item is described, for example "Cliffe Schreuders" / "Schreuders".

A complete path that locates a node is a distinguished name (DN). For example, "cn=Cliffe Schreuders,ou=academic,dc=leedsmet,dc=ac,dc=uk" could be the DN for retrieving further information about that member of staff.



LDAP example database structure

Using centralised network-based authentication via LDAP

Note that you have learned about *local authentication and user management*; for example, the use of /etc/passwd and /etc/shadow. However, in larger organisations it is often preferable to provide *network-based authentication and user management*.

Centralised identity schemes enable administrators to manage user accounts, which can be used to log into multiple networked systems within the organisation by using the same login details. This is the approach used by most medium to large organisations.

Federated identity is where user accounts from multiple organisations can be used in a connected way. For example, logging into Yahoo! with a Google account. *Single sign on (SSO)* is a related (but distinct) concept, where a user that has logged in to a service can access that service along with others, without having to log in again.

On the LinuxY 32bit openSUSE 13.1 KDE 4 desktop VM:

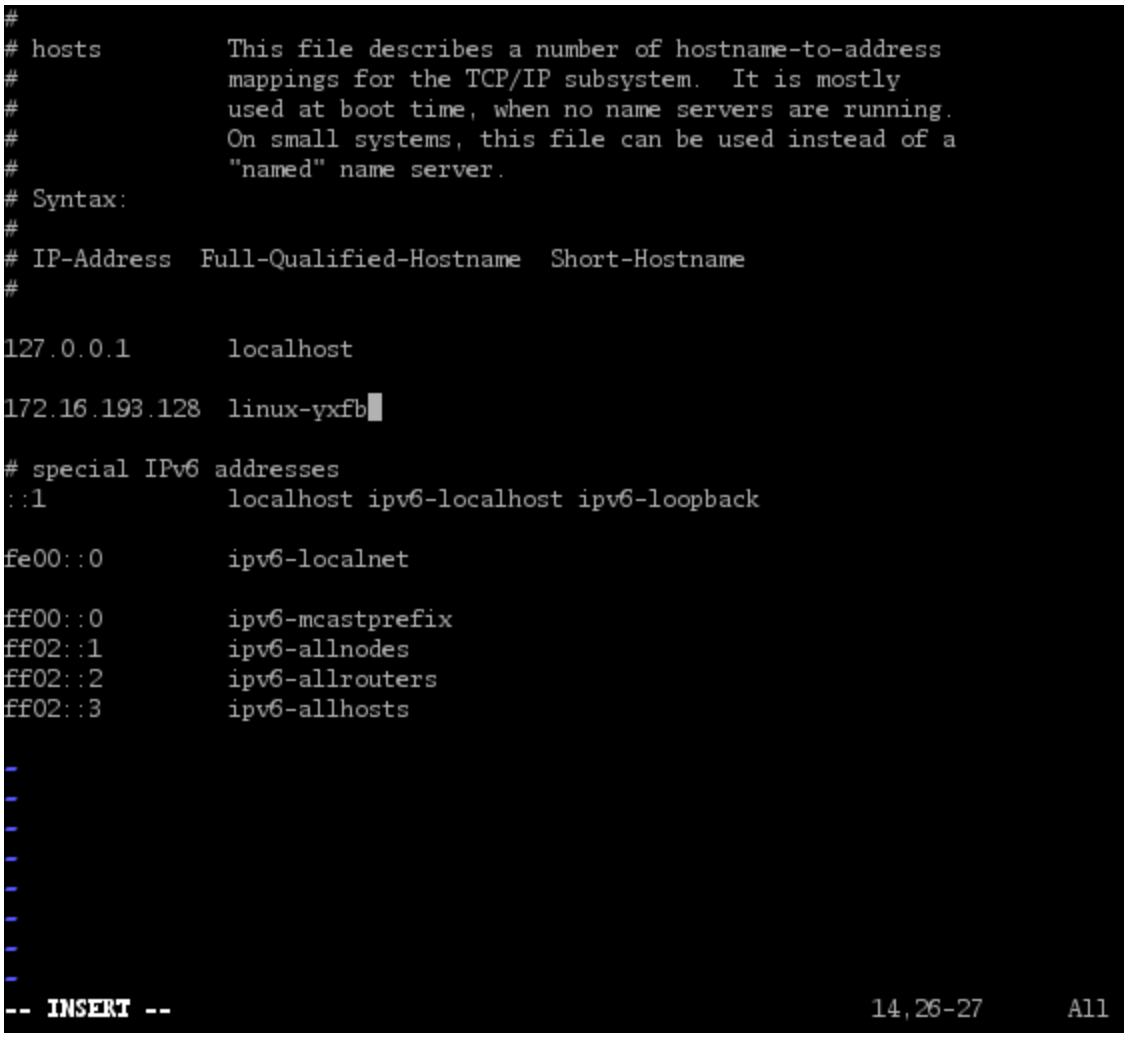
From this system you will connect to the LDAP server and configure a centralised LDAP based user management…

Start by hard-coding the name of the server on the client. This would not normally be necessary; however, this will help avoid problems with classmates using the same hostnames.

```
vi /etc/hosts
```

Add a line in the format: *ipaddress hostname*

Where *ipaddress* is the IP address of the server, and *hostname* is the hostname of the server (and the name you used when creating your server certificate). For example, “192.168.207.128 linux-yxfb”.

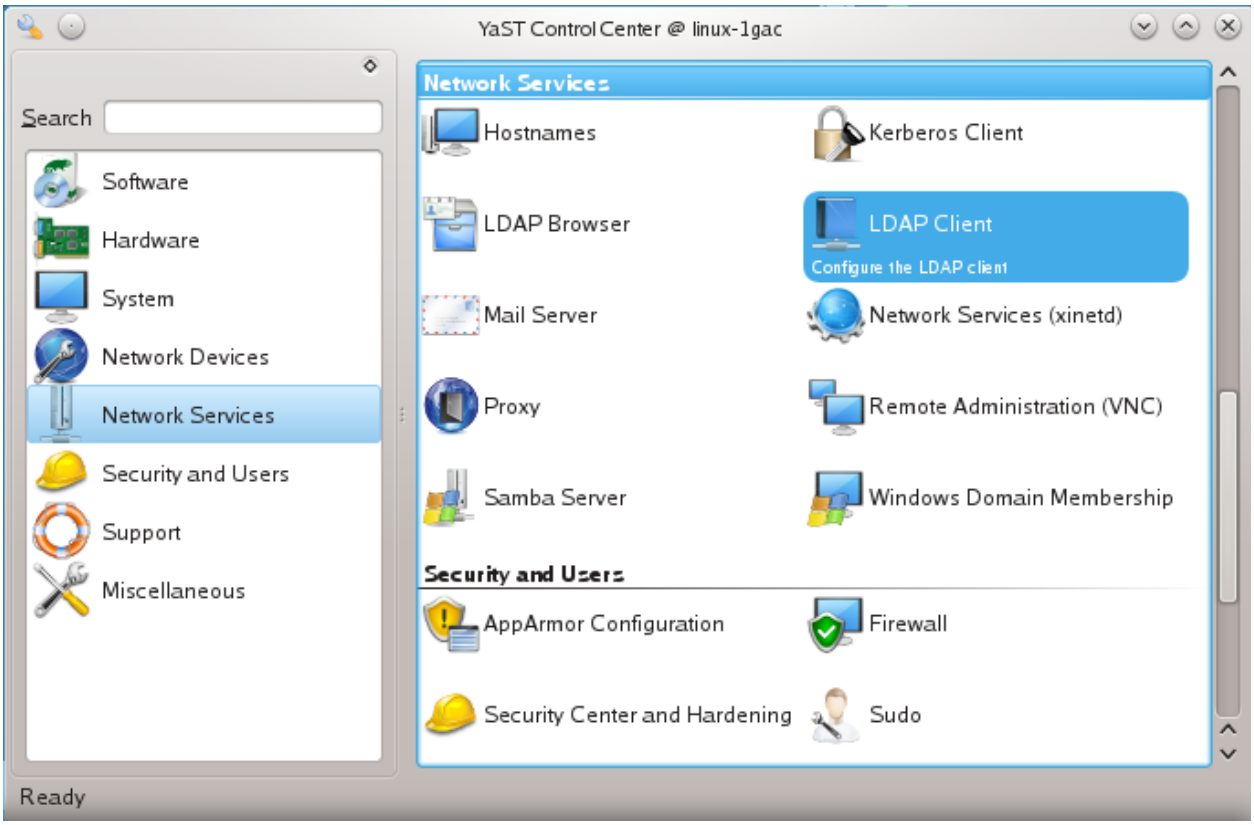


Hardcoding name resolution via /etc/hosts

Start YaST (Yet another Setup Tool), which is SUSE's configuration tool:

```
yast2
```

Navigate to "Network Services", "LDAP Client".



YaST LDAP Client module

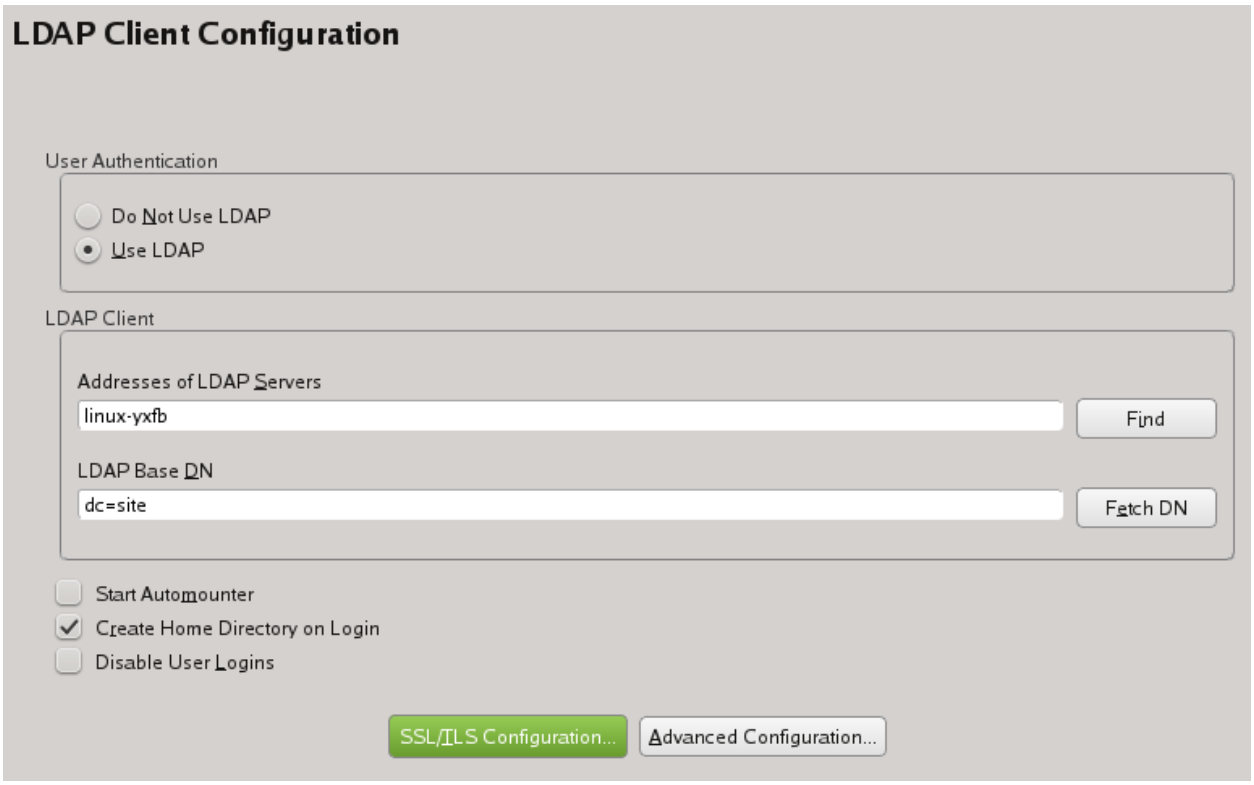
Enable “User Authentication”, “Use LDAP”.

Enter the LDAP server details:

Address of LDAP Server: *hostname* (such as linux-yxfb)

LDAP Base DN: dc=site

Check “Create Home Directory on Login”

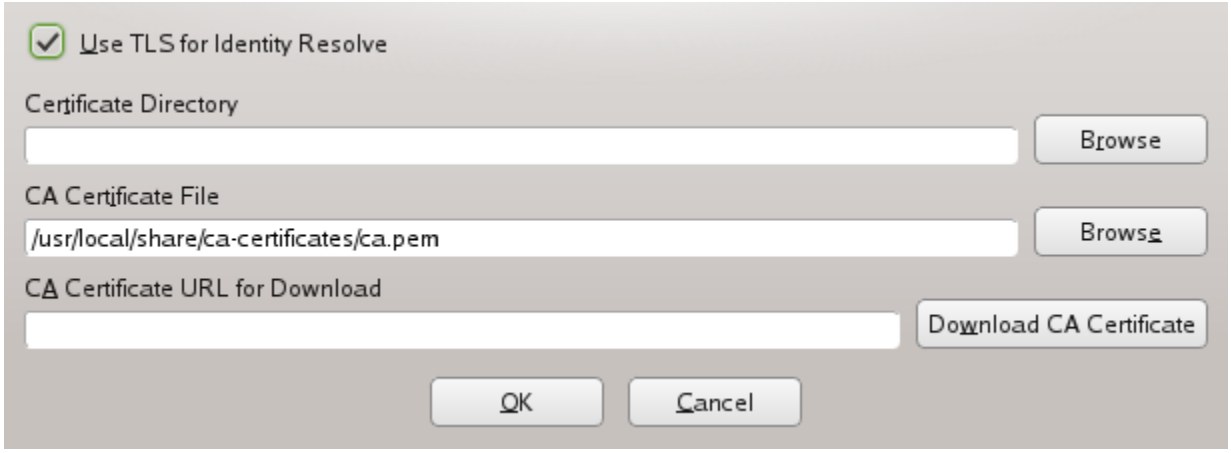


YaST LDAP client configuration, enabling authentication

Select “SSL/TLS Configuration”, to use an encrypted and authenticated connection between the client and server.

Check “Use TLS for Identity Resolve”.

Browse to the “CA Certificate File” you previously copied from the server (/usr/local/share/ca-certificates/ca.pem).



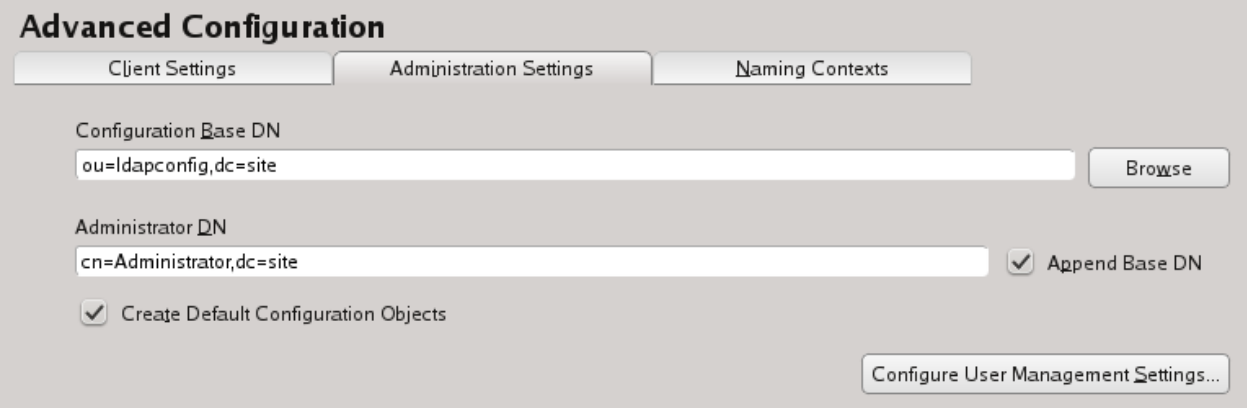
YaST LDAP client configuration, configuring encryption / CAs

Select “OK”.

Click “Advanced Configuration”.

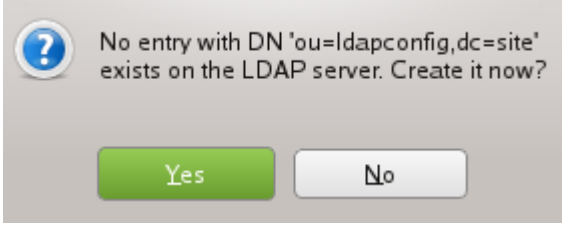
Set the “Administrator DN” to “cn=Administrator,dc=site”.

Check “Create Default Configuration Objects”, and select “Configure User Management Settings...”



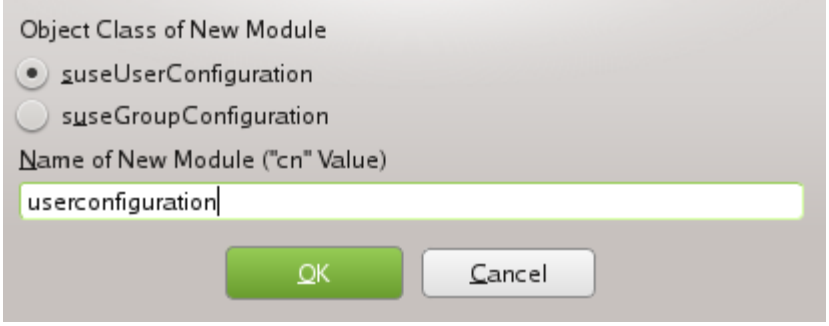
YaST LDAP client configuration, specifying LDAP the database’s DNs

Select “Yes” to create the settings on the LDAP server.



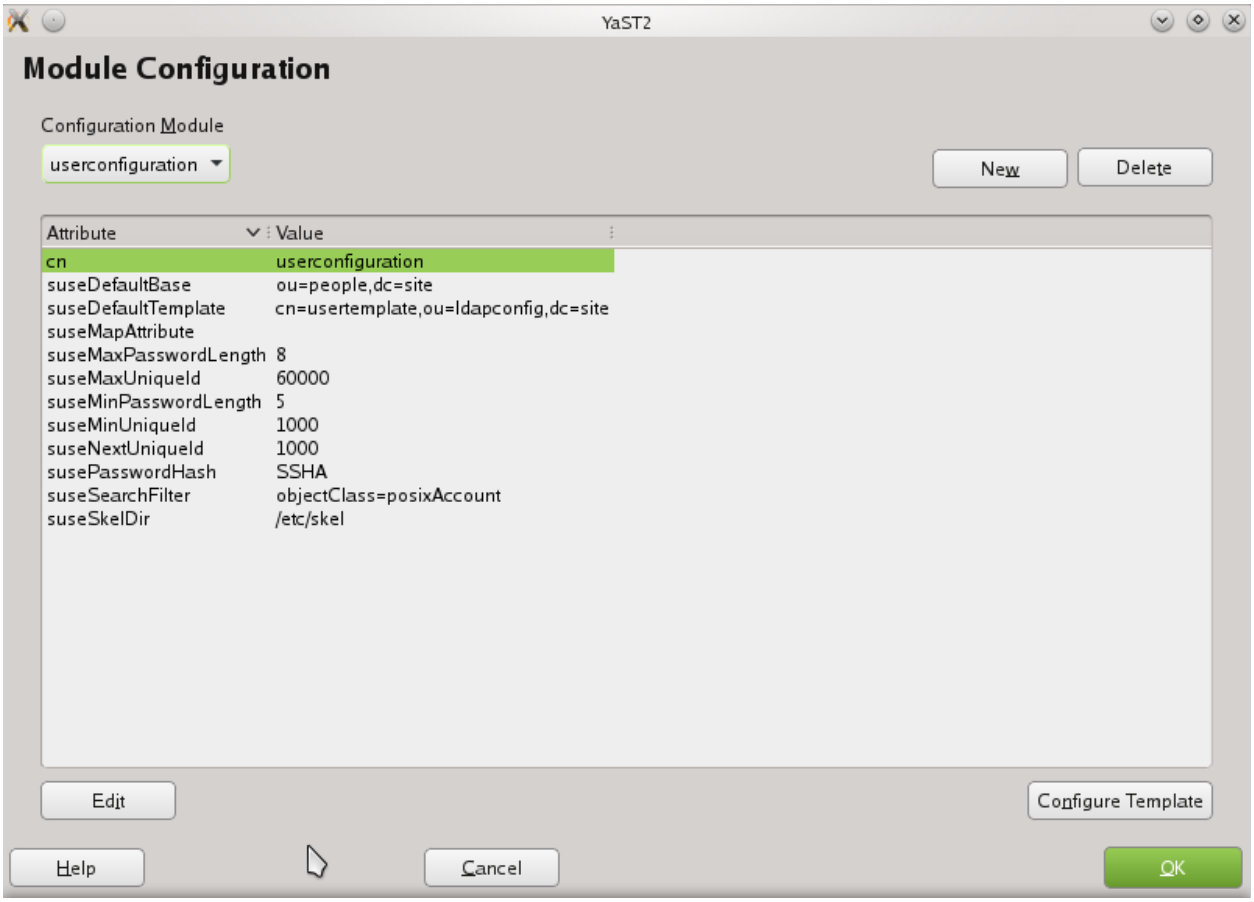
Accepting changes to the database

Click “New”, and choose “suseUserConfiguration”, and enter a name, such as “userconfiguration”.



Adding user configuration to the LDAP database

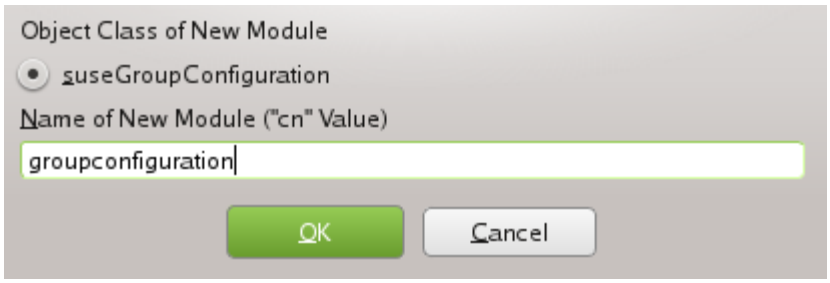
Here you can see that you can manage user account settings and password policies.



Viewing user configuration on the LDAP database

Select “Configure Template”, and look at the way the default settings for new users can be adjusted. Click “OK” to exit the Object Template Configuration

Click “New”, and do the same to add a “suseGroupConfiguration”.

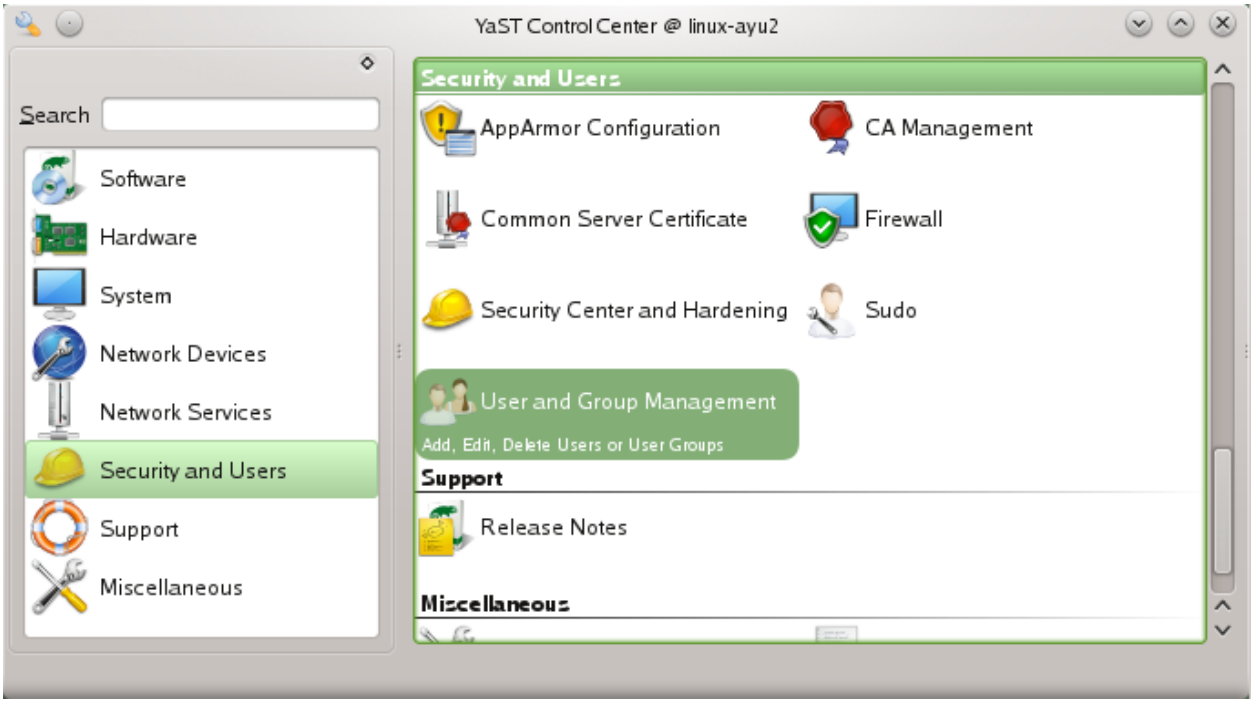


Adding group configuration to the LDAP database

Everything is now configured to authenticate via LDAP. You just need some users accounts to authenticate against.

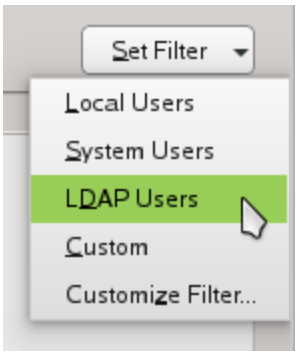
Remotely managing users via LDAP

In YaST, start “User and Group Management”.



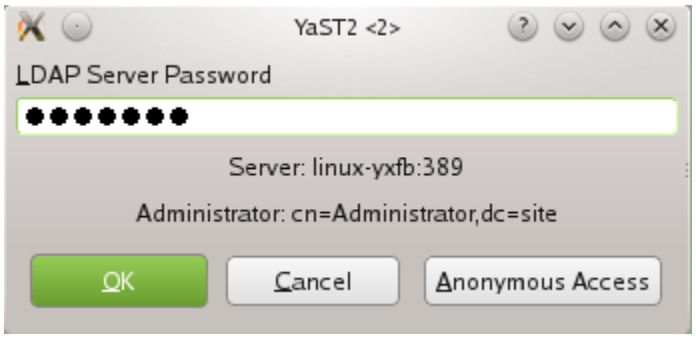
YaST User and Group Management module

Under “Set Filter”, select “LDAP Users”.



Managing LDAP users from YaST User and Group Management module

Enter the LDAP admin password that you noted earlier.

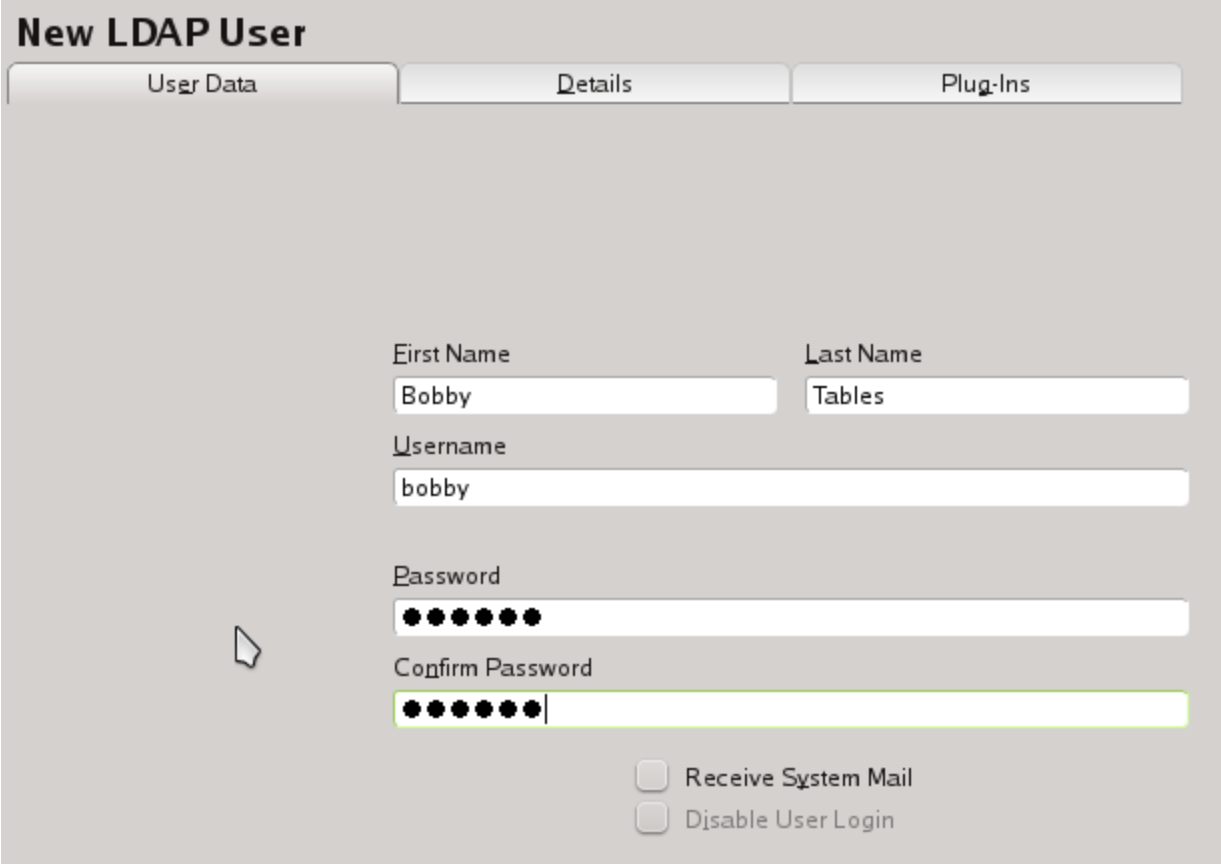


Authenticating to the LDAP server for administrative access to the database

Click “Add”, to configure a new user for LDAP authentication.

Enter a first and last name.

Enter a username and password (make a note of this username and password).

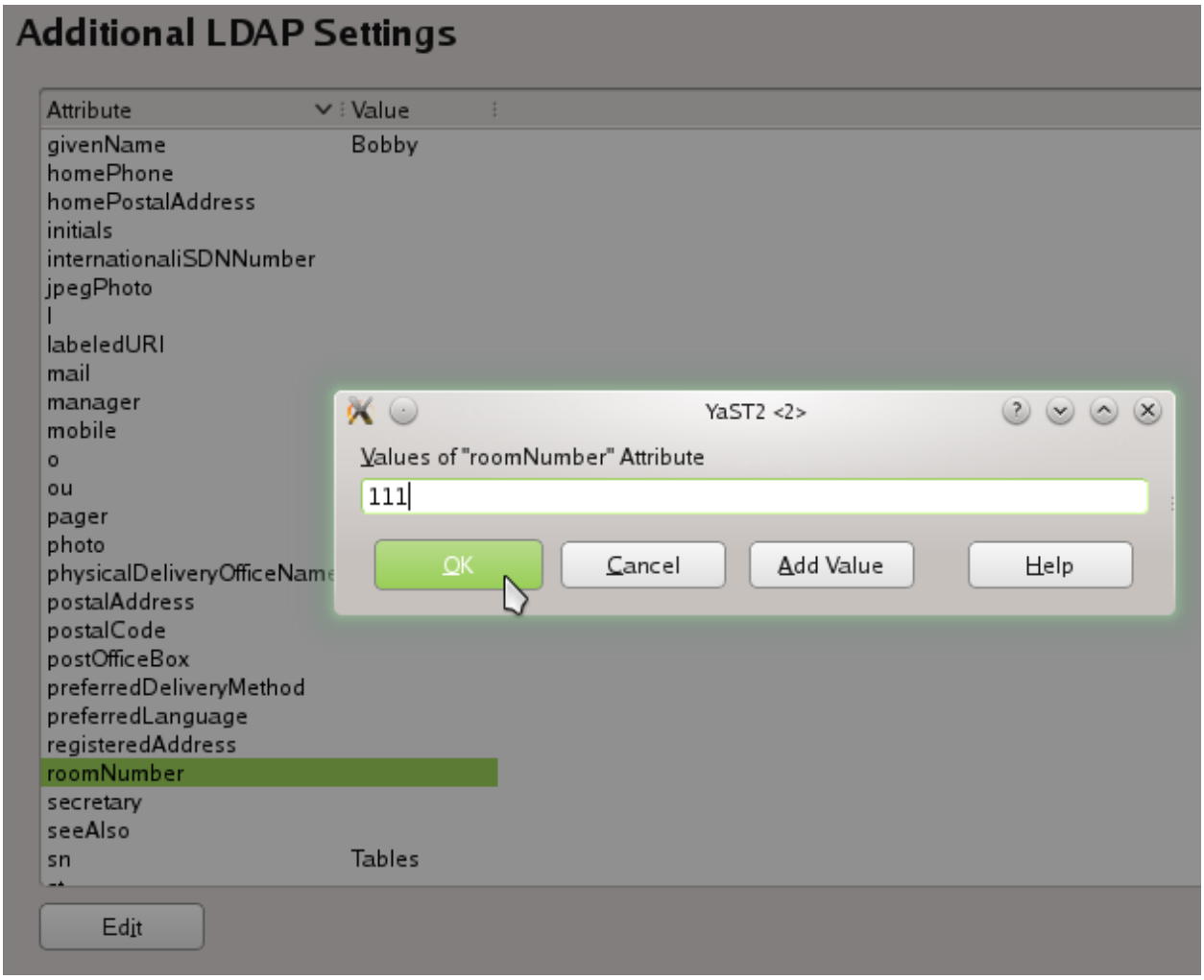


Creating a new user on the remote LDAP database

Navigate to the "Plug-Ins" Tab, and click "Launch".

As you can see, the LDAP entry for the user contains information beyond the typical Unix user account details. You can also set home phone numbers, departments, manager details, and so on.

Set a room number and some other details for the new user.



Managing extended information stored in the LDAP database about the person

Select “Ok” three times, to add the user to the LDAP database, and exit User and Group Management, back to YaST.

Logging in for the first time

Log into your new LDAP account, using the username and password you noted when creating the account. You can do this either by using the KDE Switch User feature (KDE Launch Button, Leave, Switch user), or simply from the command prompt:

```
su - username
```

or

```
ssh username@localhost
```

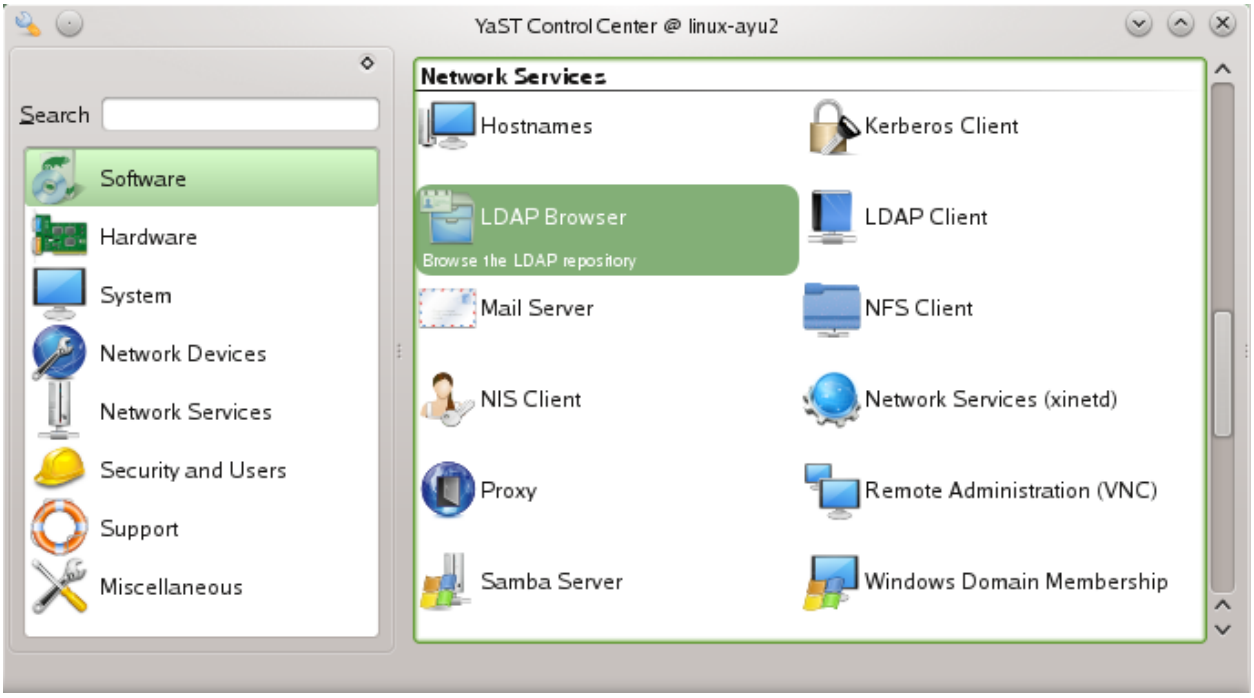
```
student@linux-ayu2:-> ssh bobby@localhost
Password:
Last login: Mon Mar 31 14:26:58 2014
Have a lot of fun...
bobby@linux-ayu2:->
```

One way to test logging in: local ssh connection

Great! You have successfully set up a centralised identity management system!

Browsing the LDAP data store

In YaST, navigate to "LDAP Browser".

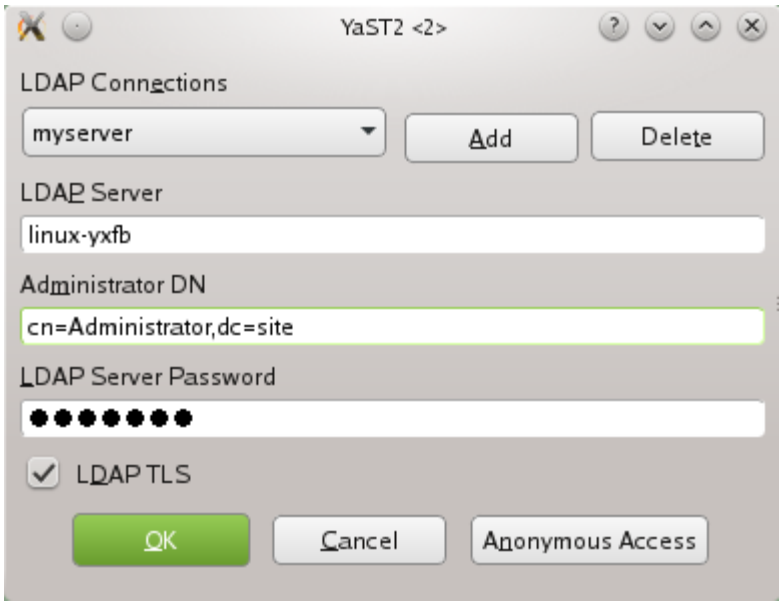


YaST LDAP Browser module

Click "Add", to setup the LDAP browsing settings, and type in a name for the connection (such as "myserver"), and accept with "Ok".

Adding a connection

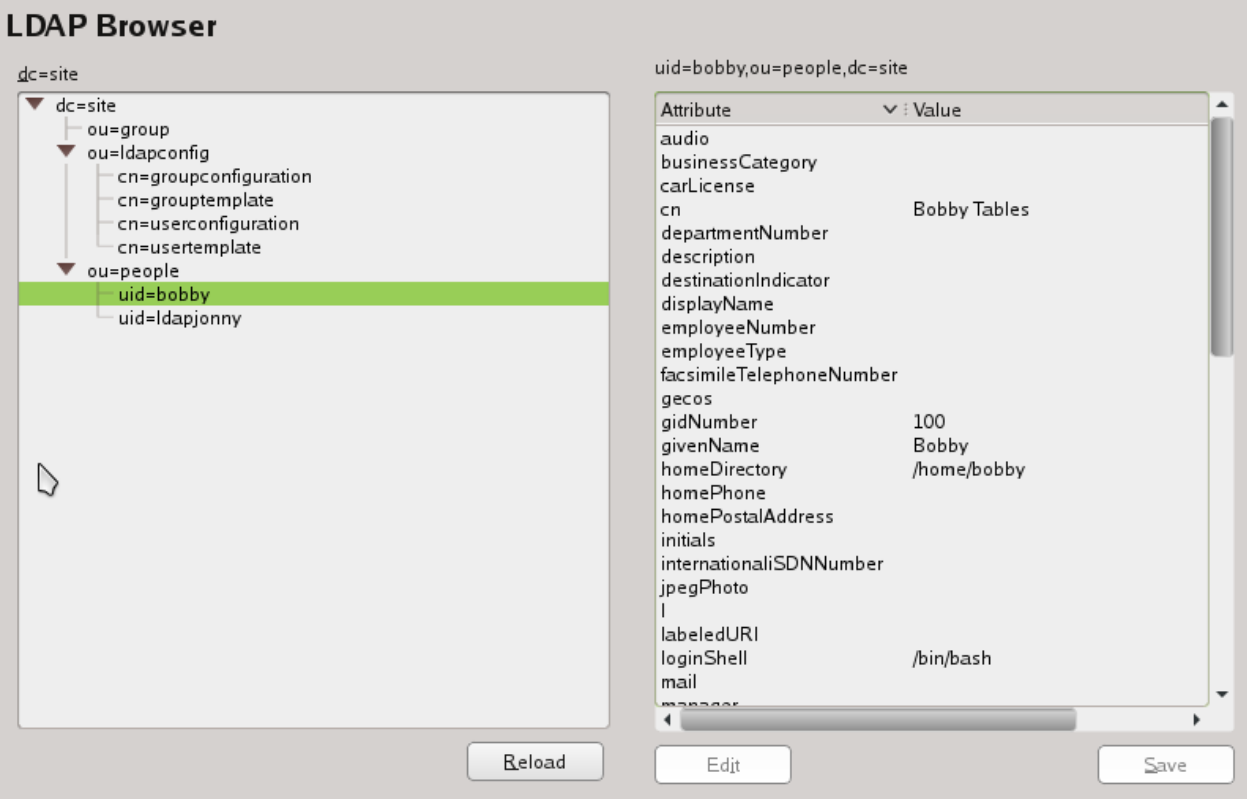
Set the Administrator DN to "cn=Administrator,dc=site", enter the LDAP admin password, check "LDAP TLS", and select "OK".



Connecting to the LDAP server from LDAP Browser

Here you can view and edit any users that have been added to the LDAP server, change user settings, and so on.

Spend some time navigating the LDAP tree and understanding how it is configured.



Viewing and editing the LDAP database via YaST LDAP Browser

Where and how is the password stored in LDAP?

Close the LDAP Browser, and restart the browser, this time login anonymously.

What kinds of information is available to anonymous (non-administrator) connections? Is the password storage visible?

Make a copy of your client VM, and then add another LDAP user account.

Answer these questions:

- Can you login to the cloned VM using the new user account created via the other VM? Why?
- Can you login to the SLES server using an LDAP user account? Why?
- What are some of the changes that YaST has made to the system to enable LDAP authentication? Does it make use of PAM?

Challenge: setup automounting of home directories, so that all Linux users have their home directories stored centrally on the SLES server.

Windows authentication via LDAP

On the WinXP Pro SP3 with pGina VM:

Windows can be extended to make use of an external LDAP authentication server. Windows includes a feature known as GINA (Graphical Identification and Authentication), which provides a way for third party vendors to add authentication methods to Windows.

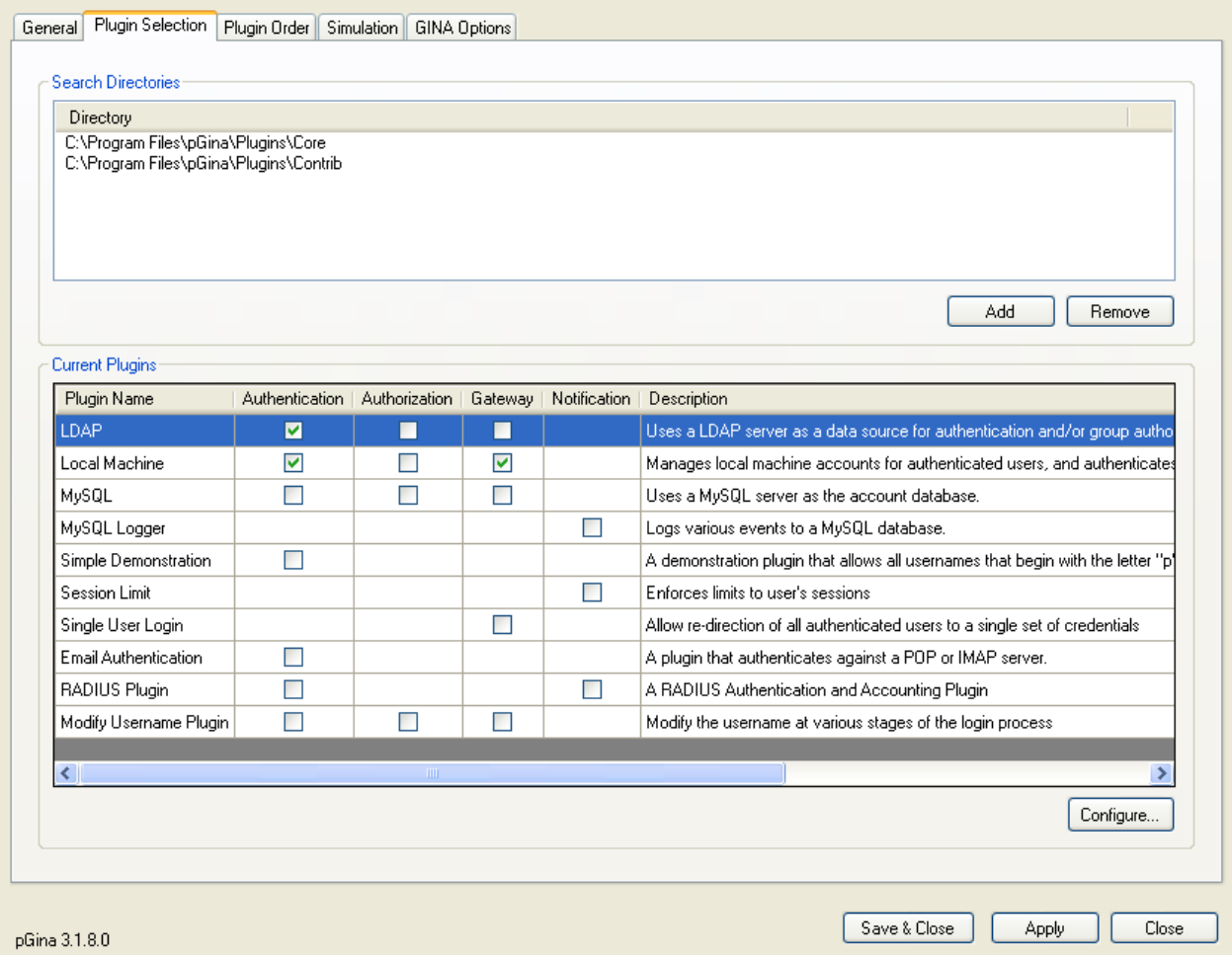
pGina makes use of GINA to extend Windows to add support for a number of alternative methods of Authentication, including LDAP.

http://pgina.org

Login to the VM using the default "Administrator" account (which currently has no password).

Start pGina (Start, All Programs, pGina, pGina).

Navigate to the "Plugin Selection" tab, and enable Authentication via LDAP.



With LDAP highlighted, click “Configure...”.

Enter the IP address of the LDAP server.

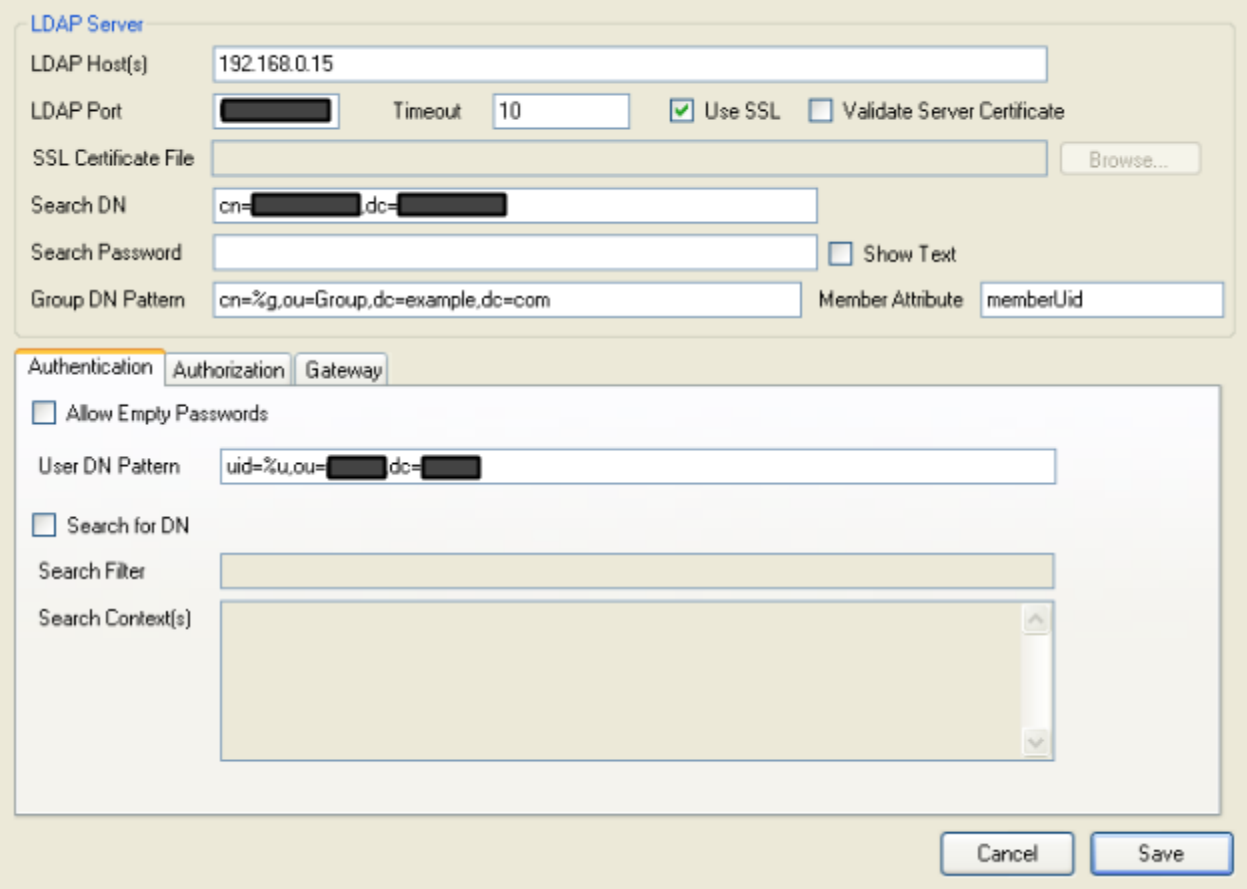
Specify the LDAP port number used for SSL connections (you will need to look this up).

Use SSL.

Do not enable “Validate Server Certificate”.

Configure pGina so it knows how to query the LDAP database to find user accounts…

> **For Search DN enter the administrator DN**, as entered previously. Hint: cn= ***********,dc= ****
>
> Under the “Authentication” tab, **set the User DN Pattern, so it can find the user accounts**. Hint: you may need to look at the contents of the LDAP database (Hint: uid=%u,ou=*p*******,dc= ****)

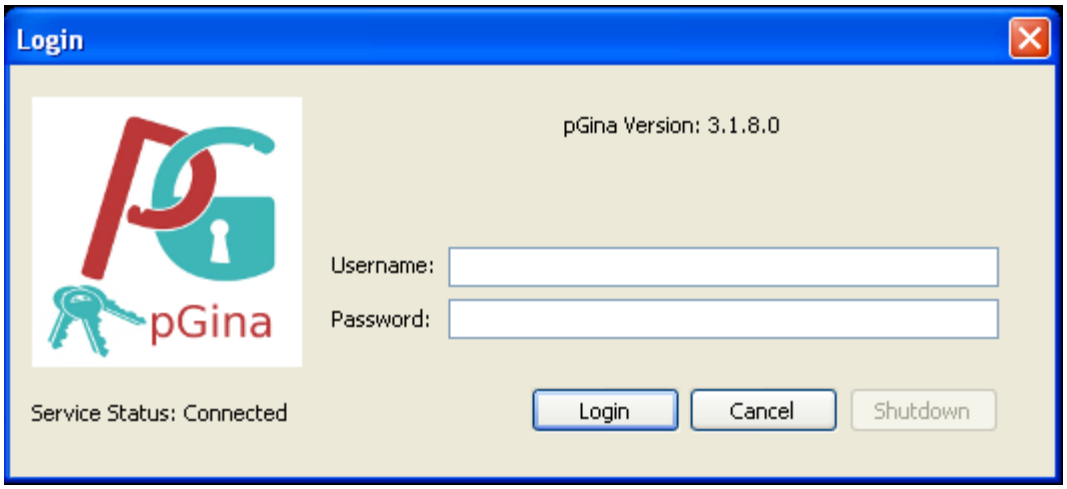


pGina, configuring the LDAP connection (hints)

Navigate to the "Simulation" tab. Under "Simulated LoginUI" enter your LDAP username and password. Check the results section and try "Show Log" to check that LDAP authentication is working.

If not, check and edit your LDAP settings until it works.

Log off from Windows, and log in with your LDAP account.



pGina, login prompt

You now have a centrally managed identity that applies across both Linux and Windows systems!

LDAP connection security: encryption, sniffing, and MITM attacks

Thanks to your use of X.509 certificates / PKI, the communications between your LDAP server and client have all been encrypted. This makes it very hard to anyone to eavesdrop on your communications.

From the LinuxZ image or from the Kali Linux VM (on LinuxZ first run "sudo chmod a+rw /dev/vmnet*"), Start Wireshark, listening on the same network as your LDAP services.

From the LinuxY 32bit openSUSE 13.1 KDE 4 desktop VM, use the LDAP browser to connect the the LDAP browser, using the administrative password.

From the VM with Wireshark, filter to the LDAP communications, and note that you cannot make sense of the encrypted traffic.

However, the Windows configuration may be subject to Man in the Middle (MITM) attacks.

What is that is insecure about this configuration, and that makes it vulnerable to MITM?

On the WinXP Pro SP3 with pGina VM:

Re-configure this connection to make it secure against MITM attacks.

Hint: you may need to copy a file across to the Windows host.

Solution:

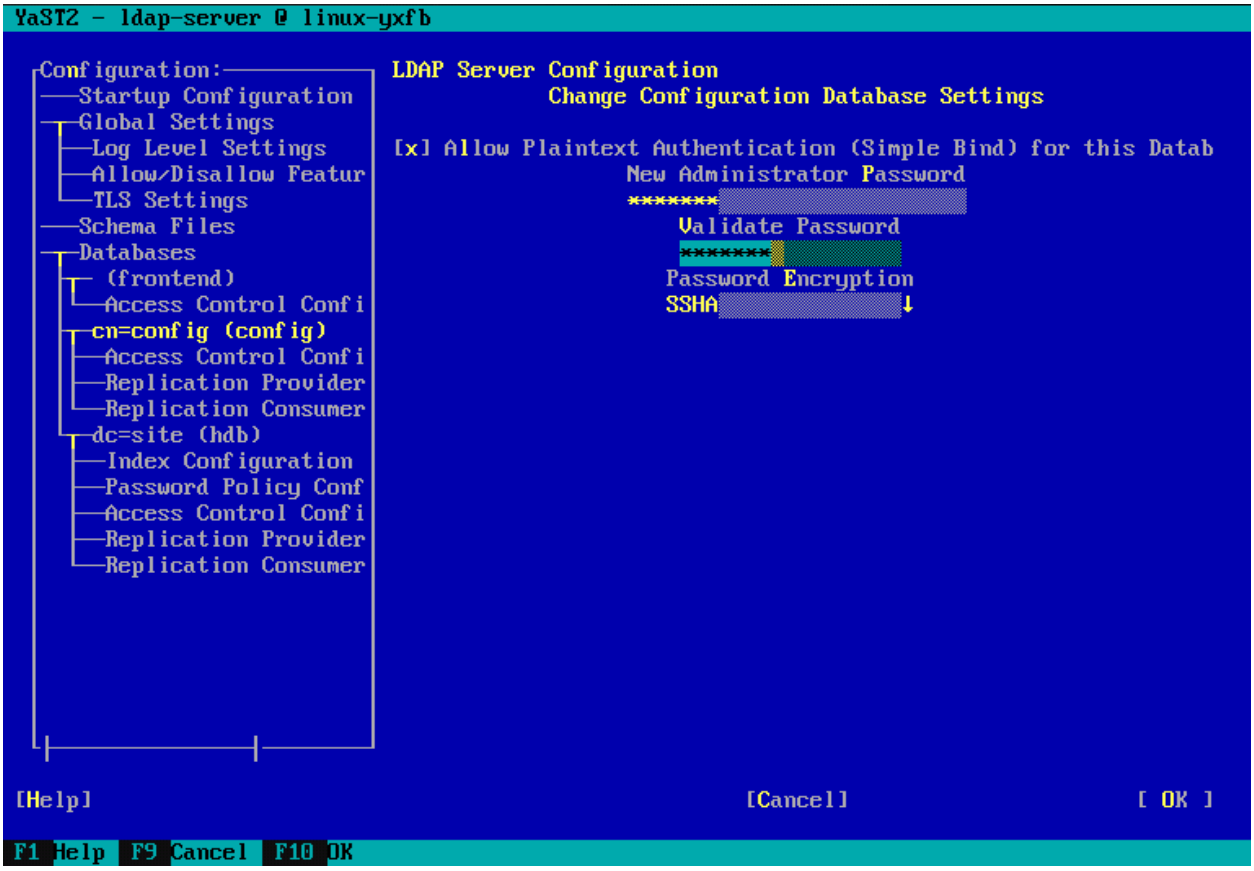
SPOILER ALERT! SPOILER ALERT! SPOILER ALERT!

You need to copy the certificate authority certificate to the Windows system, and set pGina to “Validate Server Certificate” based on that file. Otherwise an attacker on the local network may be able to MITM the connection, and pGina will ignore the fact that no trusted CA has certified the certificate matches the server.

SPOILER ALERT! SPOILER ALERT! SPOILER ALERT!

On the SUSE Linux Enterprise Server (SLES) VM:

In YaST, start the LDAP Server module, navigate to Databases,cn=config and “Allow Plaintext Authentication for this Database”.



YaST LDAP Server, enabling unencrypted connections

From the LinuxZ image or from the Kali Linux VM, Start Wireshark, listening on the same network as your LDAP services.

From the LinuxY 32bit openSUSE 13.1 KDE 4 desktop VM, use the LDAP browser to connect the the LDAP browser, using the administrative password.

From the VM with Wireshark, filter to the LDAP communications.

View the LDAP traffic in Wireshark. Can you find the password?

This illustrates the importance of using a secure connection: encrypted and verified by certificates.

Other network-based identity management solutions

LDAP is primarily a directory for storing and accessing information, and one use of LDAP is a central store of user account details. An older method of centrally storing account information is Network Information Service (NIS), which was created by Sun Microsystems. However, NIS has many security limitations (no encryption, verification, and password hashes are typically public). An enhanced version, NIS+, has been developed, but it is not very popular.

Kerberos is a network authentication protocol, which uses a trusted third party to enable clients and servers to verify each others identity, without sending passwords to each other directly. Tickets are distributed by a Key Distribution Center (KDC), which gives the authenticated client a time-limited access to the server.

Remote Authentication Dial In User Service (RADIUS) is a related protocol, which can also be used to manage authentication over a network, and is typically used to manage Wi-Fi connections and accounts.

Active Directory (AD) is Microsoft's amalgamation of LDAP and Kerberos. Windows computers are typically managed as part of a *domain*, which is implemented using a modified version of Kerberos for computers to authenticate and grant permissions to each other, and LDAP as a access protocol to retrieve information about accounts. Group policy automates specifing the security controls for Windows systems. Group policy is distributed to computers via AD automatically to any computers that are part of the same domain.

Challenge: set up an AD domain in Windows Server, and authenticate a Windows client to it.

Challenge: authenticate a Linux client to AD.

Conclusion

At this point you have:

- Created your own Certificate Authority (CA) and used it to sign a server certificate
- Used SUSE Linux Enterprise Server (SLES) and configured an LDAP server
- Managed the LDAP database remotely, adding users
- Set up a centralised network-based identity management solution, with Linux and Windows hosts all authenticating against the same accounts
- Experimented with security features such as encryption and certificate verification, to secure the communications

Well done!