HARTMAN
AUTOMATISERINGS
CONSULTANCY

# General Infrastructure

# OpenLDAP LDAPS Setup

# Red Hat Enterprise Linux 5.2

**Author: René Hartman**

**Status: For approval**

**Version: 1.2**

HARTMAN
AUTOMATISERINGS
CONSULTANCY

# DOCUMENT HISTORY

## *Document Location*

Ensure that this document is the current version. Printed documents and locally stored copies may become obsolete due to changes in the master document.

## *Revision History*

| | |
|---|---|
| Document name: | OpenLDAP_LDAPS_Setup_RHEL5.doc |
| Version: | 1.2 |
| Date of 1st version: | 22-Jan-2009 |
| Date of this version: | 17-Jun-2011 |

Note: in the following table the revision number does not necessarily match with the version number of the document. The version number of the document is merely meant to control the number of saved changes.

| **Version** | **Date** | **Author** | **Status*** | | | **Brief Description, Remarks** |
|---|---|---|---|---|---|---|
| | | | **Cpt** | **FA** | **Ap** | |
| 1.0 | 22-Jan-2009 | René Hartman | **X** | | | Initial Draft |
| 1.1 | 02-Feb-2009 | René Hartman | **X** | | | Minor mods, more uniform over RHEL versions |
| 1.2 | 17-Jun-2011 | René Hartman | **X** | | | Added remark on RHEL 5.5 configuration change |

* Cpt = Concept; FA = For Approval; Ap = Approved

HARTMAN
AUTOMATISERINGS
CONSULTANCY

## *Reviewers*

The following people have reviewed this document

| Version | Name | Role | Required date of feedback |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

## *Classifications*

Status: For approval

HARTMAN
AUTOMATISERINGS
CONSULTANCY

# TABLE OF CONTENTS


Document History ..... 1
Document Location ..... 1
Revision History ..... 1
Reviewers ..... 2
Classifications ..... 2

Table Of Contents ..... 3

Introduction ..... 4
1.1 Overview ..... 4
1.2 Document Conventions ..... 4
1.3 Prerequisites ..... 4
1.3.1 Partitions ..... 4
1.3.2 Software Repository ..... 5
1.3.3 Specific Users / Groups ..... 5

2 Client installation ..... 6
2.1 Overview ..... 6
2.2 Configuration ..... 6
2.2.1 LDAP configuration ..... 6
2.2.2 PAM configuration ..... 7
2.2.3 Caveats ..... 8
2.2.4 Upgrade woes ..... 9

3 Server installation ..... 10
3.1 Installation ..... 10
3.2 Configuration ..... 10

4 Access Policy ..... 12
4.1 Organisation: o=YOURORGANISATION ..... 12
4.2 Domains: ou=Domains ..... 12
4.3 Hosts: ou=Hosts ..... 12
4.4 Groups: ou=Group ..... 12
4.5 Users: ou=People ..... 13

5 LDAP Server synchronisation ..... 14

6 Security and contingency ..... 15

7 Maintaining LDAP with phpLDAPadmin ..... 16
7.1 Creating a new domain with phpLDAPadmin ..... 16
7.2 Creating a new host with phpLDAPadmin ..... 16
7.3 Creating a new group with phpLDAPadmin ..... 16
7.4 Creating a new account with phpLDAPadmin ..... 17
7.5 Anomalies ..... 18
Appendix A - Error codes ..... 19
Appendix B - Optimisations ..... 23

HARTMAN
AUTOMATISERINGS
CONSULTANCY

# INTRODUCTION

## 1.1 Overview

This document provides a detailed description of the installation and configuration of an LDAP(S) Client and Server.

Using LDAP(S) for user authentication creates a single point of maintenance for userids, passwords and access to servers, which has a great number of advantages over the per-box approach, among which:

- a given user always has the same uid and gid across servers
- when a user changes his/her password, this applies to all servers
- because password changes are global, there's no reason not to enforce periodic password changes
- access to servers can be granted and revoked in a central location (the user's LDAP entry)

Using LDAP for user authentication also adds a single point of failure to the login process: if for any reason the LDAP server is unavailable, users can not login to the systems. For this reason, at least two LDAP servers should be defined, which are being kept in sync wit each other. All end-user userids should then be maintained in LDAP. Only system and shared application accounts (SSA accounts) should be defined locally. Care should be taken that these SSA accounts all have identical uid and gid settings across servers.

In order for this to work, and in order to avoid having to include all SSA accounts in LDAP, PAM should be configured such that local authentication is sufficient to grant access to local users. As a fail-safe in emergency situations, root access will possible through the console, either physical or through ILO. With two or more synchronized LDAP servers, the need for this should virtually never arise.

## 1.2 Document Conventions

Each step in the installation procedure where command line user input is required is first described and then followed by the commands used.

Fonts used:

| | |
|---|---|
| Information and general guidelines | Tahoma, 10pt |
| `#/$ Commands` | `Courier New, 8pt` |
| `Command output` | `Courier New, 8pt` |

Commands are preceded by a # symbol indicating that they are executed as the root user or by a $ symbol which means they are executed as a "normal" user.

It is worth noting that when cutting and pasting commands from word documents to a putty session, occasionally a minus '-' symbol is converted to period '.' Watch out for these!

## 1.3 Prerequisites

### 1.3.1 Partitions

Some applications require that the `/tmp` partition be executable. For security reasons this directory is usually mounted 'noexec'. This is not an issue with OpenLDAP installation, but it is for starting an LDAP server, as a temporary startup script is being created in /tmp when the server is started. Mounting /tmp 'exec' is not a solution, as it leaves the system vulnerable. After installation of the server, the startup script will need to be edited. *No longer true in RHEL5.5.* This is discussed later on.

## 1.3.2 Software Repository

N/A

## 1.3.3 Specific Users / Groups

For clients, no specific user is required.

For servers, group ldap (gid 55) and user ldap (uid 55) will be created during installation of the openldap-servers package.

HARTMAN
AUTOMATISERINGS
CONSULTANCY

# 2 CLIENT INSTALLATION

## 2.1 Overview

The base logon client functionality for OpenLDAP (`openldap` and `nss_ldap`) is already part of the OS implementation, so only configuration is required. Fully configured and functional LDAP client configuration is provided when a new server is deployed using the ServerName PXE server. For systems not deployed through this mechanism manual configuration will be required. In case CLI-based LDAP search functionality is required, package openldap-clients needs to be installed additionally. This is normally not required.

## 2.2 Configuration

LDAP client configuration consists of two parts: LDAP itself and PAM.

### 2.2.1 LDAP configuration

A default RHEL5 installation installs OpenLDAP, with configuration files /etc/ldap.conf and /etc/openldap/ldap.conf. As it is confusing to have to configure two files, and (at the moment) these files support the same syntax, even though /etc/ldap.conf holds much more information than /etc/openldap/ldap.conf, we'll place a single ldap.conf file in /etc/openldap and create a symbolic link ldap.conf to it in /etc, thus making maintenance much more straightforward:

```
# mv /etc/ldap.conf /etc/openldap
# ln –s /etc/openldap/ldap.conf /etc/ldap.conf
```

On the clients, LDAP is enabled easiest through `system-config-authentication` or `authconfig`. This updates files /etc/nsswitch.conf, /etc/ldap.conf and /etc/pam.d/system-auth-ac. The graphical interface requires X to be installed. Just tick the box next to Enable LDAP Support, both on tab User Information and Authentication and specify the proper ldap server details under the button "Configure LDAP":

```
LDAP Search Base DN     : o=YourOrganisation
LDAP Server             : ldaps://< LDAP server1 >,ldaps://< LDAP server2 >
```

On tab *Options*, be sure to select "Local authorization is sufficient for local users" and "Create home directories on the first login". Leave "Use Shadow Passwords" enabled, "Password hashing algorithm" set to MD5, and the other options disabled. This configuration should be sufficient for allowing LDAP users to sign on.

As an alternative to the graphics UI, use:

```
# authconfig --update --enableldap --enableldapauth --ldapserver=ldaps://< LDAP_server1
>[,ldaps://< LDAP_server2 >] –-ldapbasedn=o=YOURORGANISATION --enablelocauthorize --
enablemkhomedir [ --nostart ]
```

The `--enablelocauthorize` and `–-enablemkhomedir` keywords modify PAM and will be discussed in the next section.

To complete the configuration and enable ssl, edit file `ldap.conf`:

Find line `#bind_policy hard` and add below it:

```
bind_policy soft
```

This will cause LDAP to fail immediately if an error occurs, rather than lock up the user's signon attempt, if something's amiss with the LDAP server.

Next, find line `#ssl start_tls` and add below it:

```
ssl on
tls_reqcert allow
```

This enforces ssl (ldaps) and will allow the use of self-signed certificates. Do *not* enable `ssl start_tls`, as that uses port 389 (ldap) to start the ssl session and thus port 389 needs to stay open. However, when leaving port 389 open, there's no way to prevent clients from using plain ldap instead of ldap-SSL. The idea is to close port 389 on the LDAP server in order to enforce SSL.

Finally, per-server authentication needs to be implemented by configuring the `pam_check_host_attr` parameter. This parameter enforces that an LDAP user will need to have this host assigned in order to be allowed to sign on to this particular host.

The resulting active lines in /etc/ldap.conf should be:

```
host ServerName.YourOrganisation.com
base o=YOURORGANISATION
uri ldaps://ServerName.YourOrganisation.com/
host ServerName.YourOrganisation.com
timelimit 120
bind_timelimit 120
bind_policy soft
idle_timelimit 3600
pam_check_host_attr yes
nss_base_passwd         ou=People,o=YOURORGANISATION?one
nss_base_shadow         ou=People,o=YOURORGANISATION?one
nss_base_group          ou=Group,o=YOURORGANISATION?one
nss_initgroups_ignoreusers
root,ldap,named,avahi,haldaemon,dbus,radvd,tomcat,radiusd,news,mailman
ssl on
tls_reqcert allow
pam_password md5
```

The URI line can hold multiple `ldaps://` specifications, separated by spaces.

## 2.2.2 PAM configuration

The "`Local authorization is sufficient for local users`" (`--enablelocauthorize`) and "`Create home directories on the first login`" (`--enablemkhomedir`) options in `system-config-authentication` (`authconfig`) configure PAM access options. They ensure that local users can still sign on and that the user's home directory structure will be created upon first login.

However, these directories will be created with the user's default umask setting, which is undesired. Therefore, file `/etc/pam.d/system-auth-ac` needs to be edited to reflect the proper umask.

In order to use pam_tally to limit the number of sign-on attempts before an account is disabled, this file also must be edited to address the pam_tally module. Do not rerun authconfig after editing this PAM file, as it will override the changes made.

A properly working RHEL5 example (valid for RHAS3 and RHAS4 as well) of `system-auth-ac` is:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth            required        pam_env.so
auth            sufficient      pam_unix.so nullok try_first_pass
auth            requisite       pam_succeed_if.so uid >= 500 quiet
auth            sufficient      pam_ldap.so use_first_pass
auth            required        pam_deny.so
auth            required        pam_tally.so magic_root deny=3

account         required        pam_unix.so broken_shadow
account         sufficient      pam_localuser.so
account         sufficient      pam_succeed_if.so uid < 500 quiet
```

```
account         [default=bad success=ok user_unknown=ignore] pam_ldap.so
account         required        pam_permit.so
account         required        pam_tally.so magic_root

password        requisite       pam_cracklib.so try_first_pass retry=3
password        sufficient      pam_unix.so md5 shadow nullok try_first_pass use_authtok
password        sufficient      pam_ldap.so use_authtok
password        required        pam_deny.so

session         optional        pam_keyinit.so revoke
session         required        pam_limits.so
session         optional        pam_mkhomedir.so skel=/etc/skel umask=0077
session         [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session         required        pam_unix.so
session         optional        pam_ldap.so
```

## 2.2.3 Caveats

A bug in:

```
nss_ldap.x86_64                         253-12.el5
nss_ldap.i386                           253-12.el5
```

causes ldap-logins (not ldap-searches) to fail w/o error; upgrade to:

```
nss_ldap.x86_64                         253-13.el5_2.1
nss_ldap.i386                           253-13.el5_2.1
```

On some installations (mainly RHAS3, but also some RHAS4) users were denied access even though everything appeared to be configured correctly. If this happens, check the following:

- account and password expiration settings in LDAP
- host assignment in LDAP
- both the uri and host directives were configured correctly in `/etc/openldap/ldap.conf`
- symlink `/etc/ldap.conf` was setup correctly
- hostname (at least short version) is defined in `/etc/hosts`

On some installations (mainly RHAS3, but also some RHAS4) PAM was found to ignore the `pam_check_host_attr` directive. As a result, users were granted access by the mere fact they were in LDAP. If this happens, verify the account settings in `/etc/pam.d/system-auth`. The line

```
account     sufficient      /lib/security/$ISA/pam_unix.so
```

seems to be responsible for this. Changing this to

```
account     required      /lib/security/$ISA/pam_unix.so
```

and adding

```
account     sufficient    /lib/security/$ISA/pam_localuser.so
```

fixes this. The added line is required to allow local access in case LDAP is down. As the `host` attribute now controls server access, the `AllowedGroups` directive in `/etc/ssh/sshd_config` can be removed, making access control dependent on a single attribute, the `host` attribute, only.

If it is desired to restrict the number of failed login attempts for a user, using pam_tally.so is required. This is achieved through adding two lines `to /etc/pam.d/system-auth-ac`:

```
auth      required    pam_tally.so magic_root deny=3
account   required    pam_tally.so magic_root
```

as the last 'auth' line, and as the last 'account' line.

When `authconfig` is run file `system-auth-ac` will be replaced, erasing those entries. So whenever it is required to rerun `authconfig`, whether the cli version, the ncurses version or the X version, these lines will need to be reinserted into the file (on older versions of RHEL, the filename is `system-auth`).

Also, `/etc/ldap.conf` is amended (ssl has been observed to be switched off, and host and uri directives are added/duplicated). These changes need to be corrected.

## 2.2.4 Upgrade woes

When upgrading RHEL5.4 to RHEL5.5, file `/etc/sysconfig/ldap`, which used to be optional, becomes required. This causes issues, depending on your setup. If this file already exists, the new file is installed as `/etc/sysconfig/ldap.rpmnew`. The LDAP service will not start then, hence the `.rpmnew` file will have to be merged with the active one.

The default content of the new 5.5 /etc/sysconfig/ldap file is:

```
SLAPD_LDAP=yes
SLAPD_LDAPS=no
SLAPD_LDAPI=no
```

This causes slapd to listen on port 389, not on port 636. For the setup documented here, the configuration should be changed to:

```
SLAPD_LDAP=no
SLAPD_LDAPS=yes
SLAPD_LDAPI=no
```

HARTMAN
AUTOMATISERINGS
CONSULTANCY

# 3 SERVER INSTALLATION

## 3.1 Installation

LDAP server functionality is not part of the standard deployment and will always have to be installed and configured manually. Obtain root acccess and enter

```
# yum install openldap-servers
```

This installs the LDAP deamon slapd, which for its configuration adds file `slapd.conf` to the `/etc/openldap` directory. For ldaps, a certificate and associated key file must be generated. An x509 certificate (valid 1826 days, or 5 years) can be generated using:

```
# openssl req -new -x509 -nodes -out /etc/pki/tls/certs/slapdcert.pem -keyout
  /etc/pki/tls/certs/slapdkey.pem -days 1826
```

The certificate (`slapdcert.pem`) should be world readable, the key (`slapdkey.pem`) should not (modes 644 and 640, respectively). Ownership for both files should be `root:ldap`.

Make sure the common name (cn) on the certificate matches the fully qualified DNS hostname for the server.

## 3.2 Configuration

`/etc/openldap/slapd.conf` should be edited to reflect the following active lines:

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/aab_nis.schema
pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args
TLSCertificateFile    /etc/pki/tls/certs/slapdcert.pem
TLSCertificateKeyFile /etc/pki/tls/certs/slapdkey.pem
access to attrs=userPassword
        by dn="cn=root,o=YOURORGANISATION" write
        by anonymous auth
        by self write
        by * none
access to attrs=shadowLastChange
        by dn="cn=root,o=YOURORGANISATION" write
        by self write
        by * read
access to dn.base="" by * read
access to *
         by dn="cn=root,o=YOURORGANISATION" write
         by * read
database        bdb
suffix          "o=YOURORGANISATION"
rootdn          "cn=root,o=YOURORGANISATION"
rootpw          {SSHA}UGJksl3bvBO9qSO6cDeBKEBk5itxEDCM
directory       /var/lib/ldap
index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
```

The encrypted rootpw value can be obtained by running `slappasswd` and entering the desired password at the prompts. `slappasswd` will then return the encrypted password string to be pasted into `slapd.conf`.

HARTMAN
AUTOMATISERINGS
CONSULTANCY

Create the non-standard schema file aab_nis.schema by copying nis.schema:

```
# cp /etc/openldap/schema/nis.schema /etc/openldap/schema/aab_nis.schema
```

Edit this file to add the host attribute to the posixAccount objectclass:

```
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount'
        DESC 'Abstraction of an account with POSIX attributes'
        SUP top AUXILIARY
        MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
        MAY ( userPassword $ loginShell $ gecos $ description $ host ) )
```

Copy file `DB_CONFIG.example` to `/var/lib/ldap as DB_CONFIG` and make ldap the owner:

```
cp /etc/openldap/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
chown ldap DB_CONFIG
```

When running on a server that has the `/mnt` filesystem mounted 'noexec', the LDAP service script needs to be edited to use `/usr/tmp`, or else LDAP can not be started. *No longer true in RHEL5.5.* Edit the start function in file `/etc/init.d/ldap` to change the temporary directory from `/tmp` to `/usr/tmp`:

```
.
.
function start() {
        configtest
        # Define a couple of local variables which we'll need. Maybe.
        user=ldap
        prog=`basename ${slapd}`
        # Build a wrapper script to exec slapd with the right arguments, to
        # avoid being tripped out by changes or weirdness in how daemon()
        # handles quoted arguments.
        wrapper=`mktemp ${TMP:-/usr/tmp}/start-slapd.XXXXXX`
        harg="ldap:///"
.
.
```

Start the LDAP server using:

```
# service ldap start.
```

Verify that the server is listening on the ldap and ldaps ports using netstat:

```
# netstat -a|grep ldap
tcp        0      0 *:ldap                      *:*                         LISTEN
tcp        0      0 *:ldaps                     *:*                         LISTEN
```

The ldap port (389) should be closed on the server's firewall to prevent non-ssl access to the server.

There's probably still a lot that can be optimized for both `slapd.conf` and `ldap.conf`, but these settings should get you a working LDAP setup with per-host access control.

# 4 ACCESS POLICY

## 4.1 Organisation: o=YOURORGANISATION

The LDAP directory will be set up with basedn o=YOURORGANISATION. Using the more common basedn dc=YourOrganisation,dc=com could easily lead to maintenance issues with all the YOUR ORGANISATIONsubdomains, that can currently not be assessed. This base entry in LDIF format, required to start the directory, looks as follows:

```
dn: o=YOURORGANISATION
objectClass: organization
objectClass: top
o: YOURORGANISATION
```

## 4.2 Domains: ou=Domains

As hosts have to belong to domains and need to be defined to be able to assign them to users, organisational unit Domains was created. This currently only holds the YourOrganisation.com domain:

```
dn: ou=Domains,o=YOURORGANISATION
objectClass: organizationalUnit
objectClass: top
ou: Domains
```

## 4.3 Hosts: ou=Hosts

All hosts that need to be signed on to interactively must be listed in the directory. Only listed hosts can be assigned to end users. If a host is not assigned to a user, that user can not sign on to that host. A host entry in LDIF format should look like the following example:

```
dn: cn=ServerName,ou=Hosts,o=YOURORGANISATION
objectClass: device
objectClass: domainRelatedObject
objectClass: top
cn: ServerName
associatedDomain: YourOrganisation.com
```

## 4.4 Groups: ou=Group

**Default group: ldap_users**

For authentication purposes, dns ou=Group, o=YOURORGANISATION and ou=People, o=YOURORGANISATION are set up. Groups will be defined under ou=Group, starting from gid=1000. Default group for all users will be cn=ldap_users, gid 1000. Additional groups will be set up as desired, where each server to be accessed will have its own group definition. A completed group entry in LDIF format should look like the following example:

```
dn: cn=ldap_users,ou=Group,o=YOURORGANISATION
objectClass: posixGroup
objectClass: top
cn: ldap_users
gidNumber: 1000
```

Access to specific servers can be based on group membership of a user for that server. However, this approach requires an LDAP server group for each server and a unique `ldap.conf` file on each server, as that has to hold the server name. For this reason it was decided to use the host attribute, even though it required a change in the nis schema. As editing of standard schemas is not recommended, nis.schema was copied to aab_nis.schema, and then edited.

LDAP groups cannot have the same name or gid as groups defined in `/etc/group`. In such a case `/etc/group` will take precedence.

## 4.5 Users: ou=People

UserIDs will be defined in LDAP under ou=People, starting at uid 1000. All LDAP users will be assigned group cn=ldap_users, gid 1000. Their objectClass will be inetOrgPerson (structural), posixAccount, shadowAccount and top. Attributes related to the shadowAccount and their settings will be as follows:

```
shadowExpire     : the user's end-of-contract date
shadowInactive   : 30 days
shadowLastChange : the date the user's password was last changed
shadowMax        : 60 days
shadowMin        : 1 day
shadowWarning    : 7 days
```

These settings will force LDAP users to change their password at least once every 60 days, require at least one day between password changes, give them a 7 day expiry warning period and allow for a 30 day grace period in which they still can sign on after password expiry. The shadowExpire attribute ensures that user's can no longer sign on after their contract was terminated. For fixed personnel this can initially be their 65th birthday or another date in the not-so-near future. For contractors this date will require maintenance as and when contracts are extended.

Other attributes to be assigned and their settings are:

```
homeDirectory: /home/<userid>
loginShell   : /bin/bash
uid          : <userid>
```

Server access is based on the host attribute; a user can only sign on to hosts that are listed in their LDAP entry.

The mail attribute is required for mail-address based authentication (e.g. for Eventum).

A completed user entry in LDIF format should look like the following example:

```
dn: cn=FirstName LastName,ou=People,o=YOURORGANISATION
sn: LastName
cn: FirstName LastName
uidNumber: 1000
gidNumber: 1000
loginShell: /bin/bash
shadowMax: 60
shadowWarning: 7
shadowExpire: 14244
mail: FirstName.LastName@YourOrganisation.com
uid: UserId
homeDirectory: /home/UserId
shadowMin: 1
shadowInactive: 30
shadowLastChange: 14152
userPassword: {CRYPT} <encrypted_password>
displayName:: FirstName LastName
givenName: FirstName
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
host: ServerName1
host: ServerName2
host: ServerName3
```

# 5 LDAP SERVER SYNCHRONISATION

This seems to be rather awkward in OpenLDAP 2.3. OpenLDAP 2.4 promises to offer much better facilities for this, so the current postion is for version 2.4 to become available for RHEL5.

Setting up a second LDAP server is relatively simple, to start off with. Synching it with the first is then just a matter of exporting the first server in LDIF format and importing the resulting file into the second one. The main issue is keeping the servers in sync afterwards.

One approach would be to define both servers to all clients, but keeping the second server down. The first server could then export its content on a daily (hourly?) basis to an LDIF file, keeping at least two generations. Should the first server fail, the second server can be started, cleared and fed with the latest LDIF of the first, losing only changes made since the last export, which, in our environment, should not be too many.

As soon as both servers are available for update, there's no telling which server will be updated.

It would probably be best to set up a small shared filesystem, e.g. in a SAN, containing a single LDAP database that's accessed by two or more LDAP servers, listed in all ldap.conf files distributed. Failover would then be transparent.

Obviously, some sort of monitoring would need to be in place to make sure that server failures are not only detected after the last server failed.

# 6 SECURITY AND CONTINGENCY

In order to tighten security it's a good idea to restrict the ability to perform the 'su' command to certain users only. A convenient way of doing this is using the pam_wheel.so module, that requires users to be either part of group wheel, or another named group. When using LDAP, there are some issues with this, that affect the ability to support the system should LDAP be down. When using group wheel, this has to be an LDAP group, not a local one, to be recognized. When using another named group, the default fallback to group 0 no longer occurs in case the group is unavailable (please refer to the PAM documentation for pam_wheel.so). As a result, this creates the risk that no users can su into root when LDAP is down, which is a less than preferable situation.

With an LDAP implementation, this issue can be remediated as follows:

- A support user can be created with uid=gid=500. This user is member of groupid 0 (root).
- Group 'wheel' can be commented out from the local group file and added to LDAP (gid=10).
- All LDAP users that need to be able to do 'su' will be member of LDAP group 'wheel'.

The net result of this is that the support user account has no meaning during normal operations (i.e. when LDAP is up); it's just an unprivileged user without any 'su' capabilities. However, once LDAP is down, group 'wheel' no longer exists and pam_wheel.so will start looking for groupid '0' membership. As the support user is part of that group, it can now be used for local sign-on and root access, until LDAP is back up and group 'wheel' membership prevails again.

# 7 MAINTAINING LDAP WITH PHPLDAPADMIN

A fairly nice LDAP browser and maintenance client is phpLDAPadmin, which can be found at http://phpldapadmin.sourceforge.net/wiki/index.php/Main_Page. Since phpLDAPadmin will not impact OS stability (it's merely a zipped archive that should be unpacked into your webroot) it is best to obtain the latest version, as it is actively developed, e.g., the RHEL5 rpm in rpm.pbone.net (`phpldapadmin-1.0.1-1.el5.noarch.rpm`) will not support the templates discussed below.

For easy maintenance with phpLDAPadmin, specific YOUR ORGANISATION templates have been created for domains (`YourOrganisationDomain.xml`), hosts (`YourOrganisationHost.xml`), groups (`YourOrganisationGroup.xml`) and accounts (`YourOrganisationAccount.xml`) that hold all the required fields for the creation of new users and which set default and lookup values as much as possible. Should more attributes be required in future, then these templates should be amended. These templates are in the creation and modification template directories under phpldapadmin, so each amendments possibly applies to two templates (except for domain, that currently only has a creation template).

NOTE: While phpLDAPadmin version 1.1.0.5 works nicely with the developed templates, it would appear that version 1.1.0.6 has a bug, as it throws an error on the first group in ou=groups (ldap_users). The other groups appear fine. This may be caused by the apparent inconsistency that all users have ldap_users as their home group but group ldap_users appears to have no members, which would indicate that the issue is caused by a bug in version 1.1.0.5. This has not yet been investigated.

For bulk inserts, it's more convenient to create an LDIF file in your favorite editor, and import that. In that case you can export a corresponding existing section of the LDAP directory first (if available) to be used as a template to edit/copy/paste from.

## 7.1 Creating a new domain with phpLDAPadmin

Select the LDAP server to update and login as cn=root,o=YOURORGANISATION.

Expand the o=YOURORGANISATION and ou=Domain entries, then select the *Create new entry here* line under ou=Domain. Select the *YOUR ORGANISATION Domain* template and fill in the domain name. As this field wil only be used for lookup of the associated domain when creating a new host, a full domainname should be entered here.

## 7.2 Creating a new host with phpLDAPadmin

Select the LDAP server to update and login as cn=root,o=YOURORGANISATION.

Expand the o=YOURORGANISATION and ou=Hosts entries, then select the *Create new entry here* line under ou=Host.

Select the *YOUR ORGANISATION Host* template, fill in the hostname and select the associated domain for this host. When setting up a new directory it's most convenient to add all hosts before creating any user accounts, as the hosts can then be set accessible when the user is being added.

## 7.3 Creating a new group with phpLDAPadmin

Select the LDAP server to update and login as cn=root,o=YOURORGANISATION.

Expand the o=YOURORGANISATION and ou=Group entries, then select the *Create new entry here* line under ou=Group.

Select the *YOUR ORGANISATION Group* template and fill in the group name. The GID Number will be assigned automatically, starting at 1000 and incrementing with each new group. The first group to be defined should be ldap_users, with GID Number 1000. This group should be created before any user accounts as it will be the main group for all LDAP users.

The default initial phpLDAPadmin UID and GID (1000) can be changed by editing config.php.

When creating a new group, existing users can simply be added to the group by selecting them in the list of userids that should become member of the group. When setting up a new directory it's therefore most convenient to create all user accounts before adding additional groups.

## 7.4 Creating a new account with phpLDAPadmin

Select the LDAP server to update and login as cn=root,o=YOURORGANISATION.

Expand the o=YOURORGANISATION and ou=People entries, then select the *Create new entry here* line under ou=People.

Select the *YOUR ORGANISATION Account* template and fill in the user details. A number of fields will be automatically populated, either based on text typed or default values. When filling in the First and Last name fields, characters containing diacritical marks should be avoided. These characters should be used for proper name display in the Display Name field, which is automatically populated from First and Last name, but should be edited when a user's name contains diacritical marks.

The User ID field is constructed from the First and Last name entries, following the YOUR ORGANISATION naming guidelines, assuming 'nl' as the user's country code. As the naming rules often get bent, this Field will most likely require editing to reflect the proper value.

The Password field should be set to the User ID, capitalized, and with '01' added to the end. E.g., user nltest should be assigned password Nltest01 initially. Encryption should be MD5. I've tried to get the template to populate the password based on the User ID automatically, but so far have not succeeded.

The UID and GID numbers are assigned automatically; UID is incremented from the last number used.

The Home directory is based on the user's userid and defaults to the standard convention `/home/<userid>`.

The Login shell defaults to `/bin/bash`; this should be valid for most, if not all, users.

The Email field is based on First and Last names and the first two characters of the UserID, which according to the YOUR ORGANISATION naming rules hold the user's country code. For most users this should be fine.

The End-of-Contract date field should be set to the user's known End-of-Contract date. This applies particularly to contractors, for permanent staff this date should be set to somewhere in the distant future. Once a (new) End-of-Contract date becomes known, this field should be amended to reflect the proper date.

The Password last changed date field should be set to the current date minus the Maximum password change interval days (default 2 months prior to the current date). This ensures that the user will be required to change his/her password upon first login.

The remaining password expiration rules are populated using defaults that should normally not be changed. Currently, PAM silently ignores the Minimum password change interval.

In the list of hosts, select all entries that the user should be able to access. If no host is selected, no sign on to any servers will be possible. This type of entry could apply to Eventum-only users, who do not need access to linux servers.

When all fields have been completed correctly, select the *Create Object* button at the bottom of the page to create the user.

## 7.5 Anomalies

Sometimes, phpLDAPadmin shows unexpected behavior, like not showing the full list of attributes properly. This mainly occurs after changing templates or the phpLDAPadmin configuration, but occasionally occurs for no apparent reason.

This can be remedied by purging the cache using the button 'Purge caches' at the top of the screen.

When templates have been modified, purging the cache is required for phpLDAPadmin to recognize the changes.

HARTMAN
AUTOMATISERINGS
CONSULTANCY

## Appendix A - Error codes

**These error messages are defined in RFC 2251 Section 4.1.10.**

| Error Name | Number | Explanation/Causes |
|---|---|---|
| LDAP_SUCCESS | 0 (x'00) | The request was successful. |
| LDAP_OPERATIONS_ERROR | 1 (x'01) | An operations error occurred. |
| LDAP_PROTOCOL_ERROR | 2 (x'02) | A protocol violation was detected. |
| LDAP_TIMELIMIT_EXCEEDED | 3 (x'03) | An LDAP time limit was exceeded. |
| LDAP_SIZELIMIT_EXCEEDED | 4 (x'04) | An LDAP size limit was exceeded. |
| LDAP_COMPARE_FALSE | 5 (x'05) | A compare operation returned false. |
| LDAP_COMPARE_TRUE | 6 (x'06) | A compare operation returned true. |
| LDAP_STRONG_AUTH_NOT_SUPPORTED | 7 (x'07) | The LDAP server does not support strong authentication. |
| LDAP_STRONG_AUTH_REQUIRED | 8 (x'08) | Strong authentication is required for the operation. |
| LDAP_PARTIAL_RESULTS | 9 (x'09) | Partial results only returned. |
| LDAP_REFERRAL | 10 (x'0A) | |
| LDAP_NO_SUCH_ATTRIBUTE | 16 (x'10) | The attribute type specified does not exist in the entry. |
| LDAP_UNDEFINED_TYPE | 17 (x'11) | The attribute type specified is invalid. |
| LDAP_INAPPROPRIATE_MATCHING | 18 (x'12) | Filter type not supported for the specified attribute. |
| LDAP_CONSTRAINT_VIOLATION | 19 (x'13) | An attribute value specified violates some constraint (e.g., a postalAddress has too many lines, or a line that is too long). |
| LDAP_TYPE_OR_VALUE_EXISTS | 20 (x'14) | An attribute type or attribute value specified already exists in the entry.<br>**Possible Causes:**<br>1. Adding entry - one or more attributes in the LDIF for an entry are exactly the same (duplicated) |

HARTMAN
AUTOMATISERINGS
CONSULTANCY

| | | |
|---|---|---|
| LDAP_INVALID_SYNTAX | 21 (x'15') | An invalid attribute value was specified. |
| LDAP_NO_SUCH_OBJECT | 32 (x'20') | The specified object does not exist in The Directory. |
| LDAP_ALIAS_PROBLEM | 33 (x'21') | An alias in The Directory points to a nonexistent entry. |
| LDAP_INVALID_DN_SYNTAX | 34 (x'22') | A syntactically invalid DN was specified. May also happen if you use an LDIF format file (dn: cn=xxx etc.) with ldapdelete which only requires a plain DN. |
| LDAP_IS_LEAF | 35 (x'23') | The object specified is a leaf. |
| LDAP_ALIAS_DEREF_PROBLEM | 36 (x'24') | A problem was encountered when dereferencing an alias. |
| LDAP_INAPPROPRIATE_AUTH | 48 (x'30') | Inappropriate authentication was specified (e.g., LDAP_AUTH_SIMPLE was specified and the entry does not have a userPassword attribute). |
| LDAP_INVALID_CREDENTIALS | 49 (x'31') | Invalid credentials were presented (e.g., the wrong password)<br>**Additional text:** unable to get TLS Client DN<br>**Possible Cause:**<br>1. No client certificate when TLSVerifyClient is demand<br>2. No client certificate when TLSVerifyClient is never in which case the error message is not fatal and service continues. |
| LDAP_INSUFFICIENT_ACCESS | 50 (x'32') | The user has insufficient access to perform the operation. |
| LDAP_BUSY | 51 (x'33') | The DSA is busy. |
| LDAP_UNAVAILABLE | 52 (x'34') | The DSA is unavailable. |
| LDAP_UNWILLING_TO_PERFORM | 53 (x'35') | The DSA is unwilling to perform the operation. **Additional text:**no global superior knowledge - the name that is being added or modified does not exist |

| | | |
|---|---|---|
| | | in any naming context or does not have a valid referral.<br>**Possible cause:** no suffix directive in slapd.conf for the DIT<br>**Additional Text:**Shadow context; no update referral - the DIT being updated is a replica in read only mode and the absence of an updateref directive means a referral cannot be returned.<br>**Possible Causes:**<br>1. A write had been attempted to a read-only replica (the consumer in a syncrepl configuration is always read-only).<br>2. In a multi-master syncrepl configuration **mirrormode true** may be missing from the slapd.conf file. |
| LDAP_LOOP_DETECT | 54 (x'36) | A loop was detected. |
| LDAP_NAMING_VIOLATION | 64 (x'40) | A naming violation occurred. |
| LDAP_OBJECT_CLASS_VIOLATION | 65 (x'41) | An object class violation occurred (e.g., a "must" attribute was missing from the entry). |
| LDAP_NOT_ALLOWED_ON_NONLEAF | 66 (x'42) | The operation is not allowed on a nonleaf object. |
| LDAP_NOT_ALLOWED_ON_RDN | 67 (x'43) | The operation is not allowed on an RDN. |
| LDAP_ALREADY_EXISTS | 68 (x'44) | The entry already exists. |
| LDAP_NO_OBJECT_CLASS_MODS | 69 (x'45) | Object class modifications are not allowed. |
| LDAP_OTHER | 80 (x'50) | An unknown error occurred. |
| LDAP_SERVER_DOWN | 81 (x'51) | The LDAP library can’t contact the LDAP server. |
| LDAP_LOCAL_ERROR | 82 (x'52) | Some local error occurred. This is usually a failed dynamic memory allocation. |
| LDAP_ENCODING_ERROR | 83 (x'53) | An error was encountered encoding parameters to send to the LDAP server. |

| | | |
|---|---|---|
| LDAP_DECODING_ERROR | 84 (x'54) | An error was encountered decoding a result from the LDAP server. |
| LDAP_TIMEOUT | 85 (x'55) | A timelimit was exceeded while waiting for a result. |
| LDAP_AUTH_UNKNOWN | 86 (x'56) | The authentication method specified to ldap_bind() is not known. |
| LDAP_FILTER_ERROR | 87 (x'57) | An invalid filter was supplied to ldap_search() (e.g., unbalanced parentheses). |
| LDAP_USER_CANCELLED | 88 (x'58) | |
| LDAP_PARAM_ERROR | 89 (x'59) | An ldap routine was called with a bad parameter. |
| LDAP_NO_MEMORY | 90 (x'5A) | An memory allocation (e.g., malloc(3) or other dynamic memory allocator) call failed in an ldap library routine. |

HARTMAN
AUTOMATISERINGS
CONSULTANCY

## Appendix B - Optimisations

This section should be amended whenever optimisations have been found and tested.

**slapd.conf**

**ldap.conf**