

LDAP Account Manager

Installation und Konfiguration

(Version 0.3)

Torsten Zumpf
torsten-z@freenet.de

12. Dezember 2006



Inhaltsverzeichnis

1	Einleitung	1
1.1	Was ist der LDAP Account Manager	1
1.1.1	Merkmale des LDAP Account Manager	1
1.2	Unterstützte Kontotypen	2
1.3	Entwickler	4
1.4	Homepage	4
1.5	Mailingliste	4
2	Allgemeines	5
3	LDAP Account Manager	7
3.1	Installation und Konfiguration	7
3.1.1	Installation	7
3.1.2	Konfiguration	9
3.2	Benutzer und Gruppen anlegen	14
4	lamdaemon	15
4.1	Konfiguration	15
4.1.1	Administrator anlegen	16
4.1.2	Einstellungen auf dem LAM-Server	17
4.1.3	Einstellungen auf dem Homeverzeichnis-Server	17
4.2	Testen des lamdaemon	19
5	Apache 2	21
5.1	Installation von Apache 2	21
6	LDAP	23
6.1	Installation und Konfiguration des Servers	23
6.1.1	Installation	23
6.1.2	Konfiguration	24
6.2	Installation und Konfiguration des Clients	28
6.2.1	Installation	29
6.2.2	Konfiguration	29
6.2.3	Konfigurieren von PAM	31

7	Verschlüsselung mit SSL/TLS	33
7.1	CA Zertifikat und Schlüssel	34
7.2	Server-Zertifikat und Schlüssel	35
7.2.1	Zertifikat signieren	36
7.3	Zertifikat einrichten	36
7.4	SSL und Apache 2	37
7.5	SSL und LDAP	39
7.5.1	SSL auf dem LDAP-Client	40

1 Einleitung

Für diese Dokumentation wurde der LDAP Account Manager in Version 1.1.1, Debian 3.1/Sarge, Apache 2 und OpenLDAP verwendet. Es wird davon ausgegangen, dass eine laufende Debian-Grundinstallation vorhanden ist. Nach einigen allgemeinen Worten wird auf die Installation und Konfiguration des LAM eingegangen. Im Anschluss daran finden sich Erklärungen zur Installation und Konfiguration der benötigten weiteren Dienste.

Noch etwas in eigener Sache. Diese Anleitung ist noch nicht vollständig und ich arbeite weiter daran. Leider schleichen sich immer mal Fehler ein, die man selber nicht mehr sieht. Gerne nehme ich Hinweise zu deren Beseitigung entgegen.

1.1 Was ist der LDAP Account Manager

Der LDAP Account Manager ist ein Web-Frontend, mit dem sich die Einträge in einem LDAP-Server verwalten lassen. Der LAM verwaltet Benutzer-, Gruppen- und Host-Accounts für Unix und Samba. Er kann Homeverzeichnisse anlegen und löschen, sowie Benutzer- oder Gruppenquotas anlegen. Die Benutzung ist teilweise selbsterklärend, jedoch sollte Basiswissen zu LDAP vorhanden sein.

1.1.1 Merkmale des LDAP Account Manager

In einer kurzen Übersicht sollen die Merkmale des LAM dargestellt werden:

- Verwaltung von Unix Benutzer- und Gruppenkonten (posixAccount/posixGroup)
- Verwaltung von Samba 2.x/3 Benutzer- und Hostkonten (sambaAccount/sambaSamAccount)
- Verwaltung von Kolab 2 Konten (kolabInetorgPerson)
- Profile für die Erstellung von Konten
- Kontenerstellung mittels hochladen einer Datei
- Automatisches Erstellen und Löschen von Home-Verzeichnissen
- Einstellen von Quotas

- Ausgabe von PDFs für alle Konten
- Editor für “organizational units” (ou)
- Schemaansicht
- Baumansicht
- Verschiedene Konfigurationsdateien
- Unterstützung von LDAP+SSL
- Unterstützung verschiedener Sprachen
 - Katalanisch
 - Traditionelles Chinesisch
 - Holländisch
 - Englisch
 - Französisch
 - Deutsch
 - Ungarisch
 - Italienisch
 - Japanisch
 - Russisch
 - Spanisch

1.2 Unterstützte Kontotypen

Unix

Typ: users und groups

Objektklassen: posixAccount, shadowAccount, posixGroup

Schema: nis.schema

Samba 3

Typ: users, groups, hosts, domains

Objektklassen: sambaSamAccount, sambaGroupMapping, sambaDomain

Schema: samba.schema

Samba 2

Typ: users und groups

Objektklassen: sambaAccount

Schema: samba.schema

Kolab 2

Typ: users

Objektklassen: kolabInetOrgPerson

Schema: kolab2.schema, rfc2739.schema

Address book entries

Typ: users

Objektklassen: inetOrgPerson

Schema: inetorgperson.schema

Mail routing

Typ: users

Objektklassen: inetLocalMailRecipient

Schema: misc.schema

Mail aliases

Typ: users

Objektklassen: nisMailAlias

Schema: misc.schema

MAC addresses

Typ: hosts

Objektklassen: ieee802device

Schema: nis.schema

Simple Accounts

Typ: users

Objektklassen: account

Schema: cosine.schema

SSH keys (LPK patch)

Typ: users

Objektklassen: ldapPublicKey

Schema: openssh-lpk.schema

1.3 Entwickler

LAM wurde und wird entwickelt von Michael Duerchner, Roland Gruber und Tilo Lutz.

1.4 Homepage

Die Webseite des LAM-Projektes findet man unter folgender URL:

<http://lam.sourceforge.net/index.htm>

1.5 Mailingliste

Es gibt zwei Mailinglisten. Zum einen eine für aktuelle Nachrichten der Entwickler (*lam-announce*) und eine allgemeine für Fragen, Erfahrungen und Diskussionen (*lam-public*).

Anmelden kann man sich auf dieser Seite:

<http://lam.sourceforge.net/maillinglists/index.htm>

2 Allgemeines

Um den LDAP Account Manager zu nutzen, ist es nötig, den Webserver Apache zu installieren. Vorzugsweise sollte dieser in Version 2 vorhanden sein. Mit Apache 1.3 läuft der LAM derzeit ebenfalls, jedoch wird irgendwann der Support dafür eingestellt. Hinweise zur Installation des Apache 2 finden sich in Kapitel 5.

Weiterhin werden verschiedene PHP-Module für den Apache benötigt. In der vorliegenden Installationsvariante und Nutzung des LAM kommt PHP 4 zum Einsatz. Eine Umstellung auf PHP 5 wird erfolgen, sobald Debian/etch Stable wird. Über die Mailingliste des LAM wurde die Einstellung des Supports für PHP 4 angekündigt. Da viele Nutzer des LAM jedoch aus verschiedenen Gründen ältere, stabile Linuxinstallationen haben und zum Teil auch laufen lassen müssen, kann damit gerechnet werden, dass PHP 4 noch eine Weile unterstützt wird. Auf die Installation von PHP wird ebenfalls in Kapitel 5 eingegangen.

Auf eine vorhandene OpenLDAP-Installation kann man ohne Probleme mit dem LAM zugreifen. Muss LDAP neu installiert werden, was man vor der ersten Aktivierung des LAM machen sollte, findet man Hinweisen im Kapitel 6. Die OpenLDAP-Installation kann auf dem gleichen Rechner erfolgen, auf dem der LAM (mit Apache) installiert ist. Aus Sicherheitsüberlegungen ist es aber ratsam, LDAP einen eigenen Rechner zur Verfügung zu stellen.

Ebenso ist es möglich, die Home-Verzeichnisse der Benutzer, welche mit LDAP verwaltet werden, auf dem gleichen Rechner wie den LAM und/oder OpenLDAP einzurichten. Sie dürfen aber auch auf einem anderen Rechner liegen.

Bei Verteilung auf verschiedene Rechner, muss für eine sichere Übertragung im Netz gesorgt werden. Für diesen Zweck wird SSL genutzt. Dazu mehr im Kapitel 7.

Für diese Anleitung sind die Dienste auf verschiedene Rechner verteilt. Die verschiedenen Schritte der Installation lassen sich jedoch auch auf einen einzelnen Rechner anwenden.

In Tabelle 2.1 wird die Übersicht der verwendeten Server und der darauf laufenden Dienste dargestellt.

Dienst	Server-Name
Apache2	ServerA
LAM	ServerA
OpenLDAP	ServerB
Home-Verzeichnisse	ServerC

Tabelle 2.1: Aufteilung der Dienste auf die Server

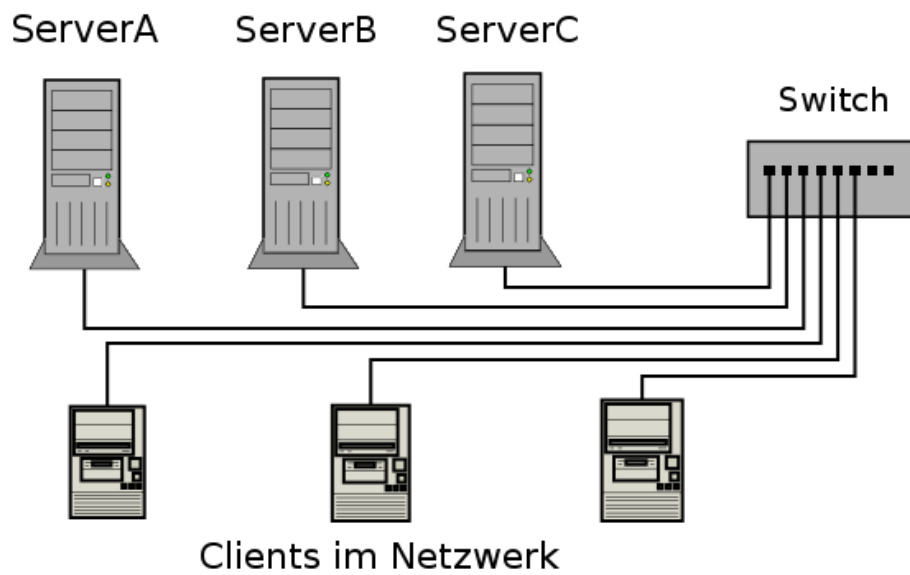


Abbildung 2.1: Netzwerkübersicht

Abbildung 2.1 zeigt vereinfacht das vorhandene Netzwerk. Die Anbindung ans Internet, welche vorhanden ist, wird dabei nicht berücksichtigt.

3 LDAP Account Manager

Im Abschnitt 3.1 wird die Installation und Konfiguration des LAM erklärt. In den darauf folgenden Abschnitten (ab 3.2) wird beschrieben, wie Benutzer, Gruppen und anderes im LDAP-Verzeichnis angelegt und verwaltet werden. Ist noch keine LDAP- und/oder Apache-Installation vorhanden, sollten zuerst die Kapitel 5 und 6 durchgearbeitet werden. Der LDAP Account Manager kann ebenso für ein schon vorhandenes LDAP-Verzeichnis Verwendung finden.

Für die Installation des LAM gibt es fertige Debian- und RPM-Pakete. Hier wird die Installation unter Debian erklärt. Bei RPM-basierten Systemen sollte es ebenso umsetzbar sein, kann sich aber in einigen Punkten (Paketbenennung, ...) unterscheiden. Auf der LAM Webseite werden fertige deb- und rpm-Pakete zum Download angeboten. Ist ein eigener Paketserver vorhanden (z.B. apt-proxy unter Debian), kann dieser genutzt werden. Ebenso befindet sich die aktuelle Version des LAM in *Debian/Sid*.

3.1 Installation und Konfiguration

Sind alle zusätzlich benötigten Pakete auf dem Server mit der Apache-Installation installiert und konfiguriert, gestaltet sich die Installation des LAM einfach.

3.1.1 Installation

Bezieht man das Paket aus *Debian/Sid* oder von einem eigenen Paketserver, installiert man den LAM mit folgendem Befehl:

```
root@debian:~# apt-get install ldap-account-manager
```

Hat man nur das deb-File geladen, erfolgt die Installation mit dem Aufruf:

```
root@debian:~# dpkg -i ldap-account-manager_1.1.1-1_all.deb
```

Unter Umständen, wenn man den Apache bereits installiert hat, werden weitere Pakete nachgezogen. Kommt es bei der Installation des deb-Files mittels dpkg zu einer Fehlermeldung, so kann mit dem Befehl

```
root@debian:~# apt-get -f install
```

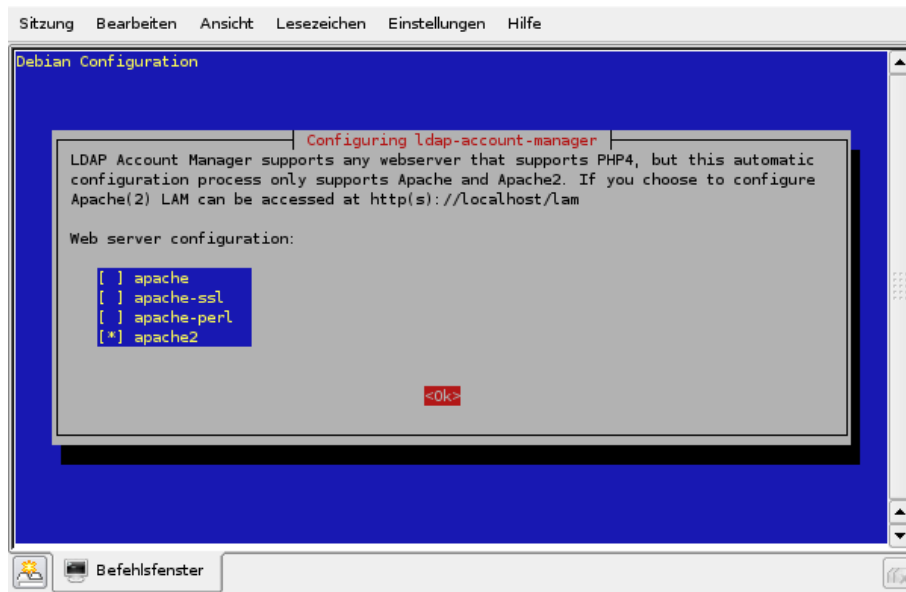


Abbildung 3.1: Apache-Konfiguration

die Auflösung der Abhängigkeiten und die Fertigstellung der Installation erreicht werden. Der Apache-Webserver wird während der Installation entsprechend konfiguriert. Welche Version Verwendung findet, kann, wie in Abbildung 3.1 dargestellt, ausgewählt werden. Anschließend wird gefragt (siehe Abbildung 3.2), ob ein Neustart des Apache-Webservers erfolgen soll. Dies ist nötig, damit die Änderungen eingelesen werden. Damit ist die Installation des LDAP Account Manager abgeschlossen.

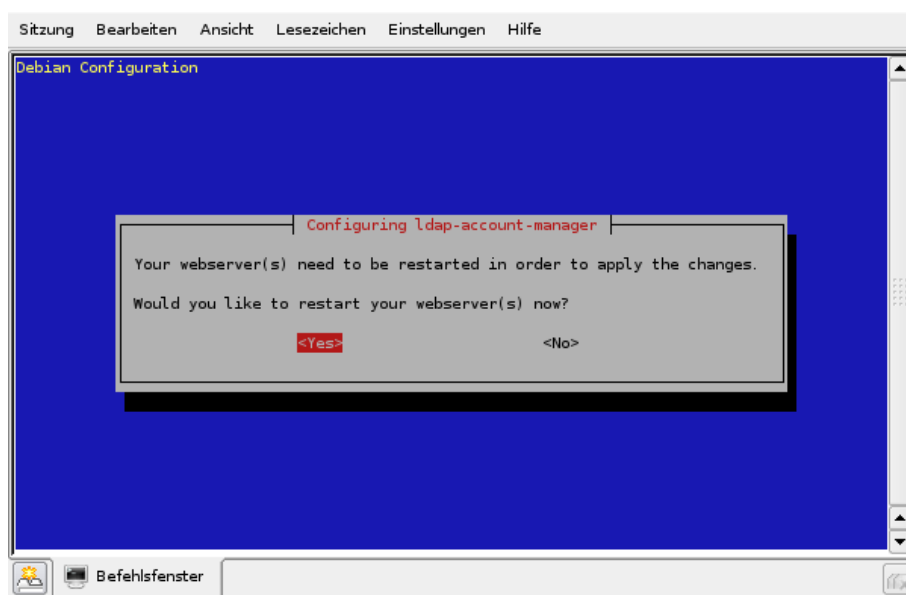


Abbildung 3.2: Neustart des Apache-Webservers

3.1.2 Konfiguration

Nach Abschluss der Installation kann der LAM über einen Browser aufgerufen werden. Wird dies lokal gemacht, ruft man `http://localhost/lam` auf. Läuft der Apache-Webserver auf einem anderen Rechner gibt man entsprechend den Namen des Servers an: `http://servername/lam`. Damit gelangt man zum Login des LAM, wie in Abbildung 3.3 zu sehen ist. Die voreingestellte Sprache ist Englisch, daher werden alle folgendes Bildschirmschnappschüsse in Englisch sein, bis die noch durchzuführende Umstellung auf Deutsch aktiv wird.

Auf der LDAP Account Manager-Startseite wählt man den Punkt **LAM configuration**. Es werden zuerst die allgemeinen Einstellungen gemacht. Diese erreicht man über den Menüpunkt **Edit general settings** (siehe Abbildung 3.4). Nun wird man aufgefordert, dass Passwort für den LAM einzugeben. Dieses lautet nach der Erstinstallation *lam*.

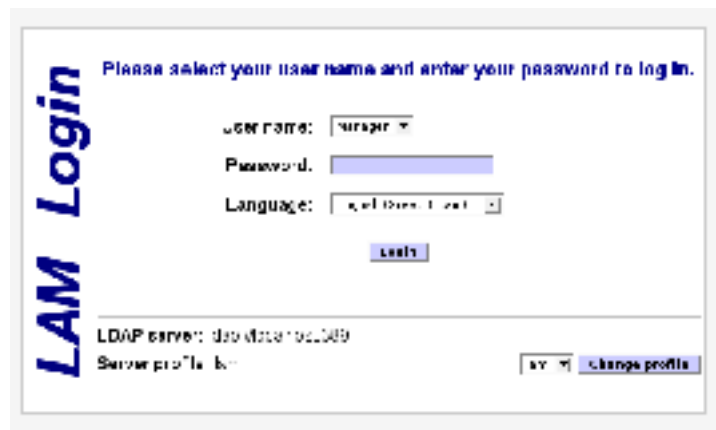


Abbildung 3.3: LAM – erster Login

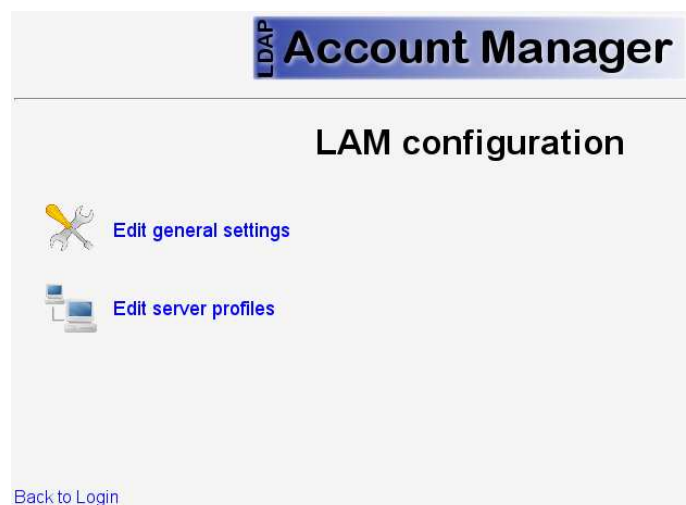


Abbildung 3.4: LAM – Konfigurationsmenü

The screenshot shows the configuration interface for the LDAP Account Manager administrator. It is divided into three main sections:

- Security settings:** Contains a 'Session timeout' dropdown menu set to '30' and an empty 'Allowed hosts' text area. Both fields have a blue question mark icon to their right.
- Logging:** Contains a 'Log level' dropdown menu set to 'Notice' with a blue question mark icon. Below it, the 'Log destination' is set to 'System logging' (indicated by a checked radio button), with 'No logging' and 'File' as other options. The 'File' option has an empty text input field next to it.
- Change master password:** Contains two red text labels: 'New master password' and 'Reenter new master password', each followed by a blue question mark icon and an empty text input field.

An 'ok' button is located at the bottom left of the configuration area.

Abbildung 3.5: LAM – Konfiguration des Administrators

Auf der folgenden Seite (Abbildung 3.5) kann man einstellen, nach wieviel Minuten der LAM einen Zwangslogout vornehmen soll. Ebenso ist es hier möglich IP-Adressen von Hosts anzugeben, welchen der Zugriff auf den LAM erlaubt sein soll¹. Weiterhin können Einstellungen zum Loggingverhalten gemacht werden. Wichtig ist der letzte Punkt. Hier sollte ein langes Passwort vergeben werden, welches auch Sonderzeichen enthält. Dies ist das Masterpasswort für den LAM. Sind die Einstellungen gemacht, landet man wieder auf der Login-Seite (siehe Abbildung 3.3).

3.1.2.1 LAM-Profile

Als nächstes muss man ein Administrationsprofil für den LDAP Account Manager einrichten. Der LAM kann von mehreren Administratoren bedient werden. Ein Hauptadministrator erstellt

¹Zugriffsregeln über IP-Adressen stellen keinen sehr sicheren Schutz dar, da sich IP-Adressen fälschen lassen.

die Profile. Um das Standardprofil (*lam*) zu bearbeiten, wird wieder der Punkt **LAM configuration** ausgewählt und dann **Edit server profiles**. Das hier einzugebende Passwort ist unabhängig von dem bereits geänderten (Masterpasswort) und lautet nach der Erstinstallation ebenfalls *lam*. Auf dieser Seite werden die, folgend beschriebenen, Einstellungen für das Standardprofil vorgenommen.

Server settings – Abbildung 3.6 In diesem Bereich gibt man bei `Server address` den Namen des LDAP-Servers an. Lauscht der LDAP-Server auf dem Standardport 389, so kann die Angabe des Ports entfallen. Bei `Tree suffix` wird die oberste Bezeichnung des LDAP-Baumes eingetragen (z.B. `dc=yourdomain,dc=org`). Um die Kommunikation mit dem LDAP-Server über SSL laufen zu lassen (und damit über den Port 636), sollte im Kapitel 7 ab Seite 33 nachgelesen werden.

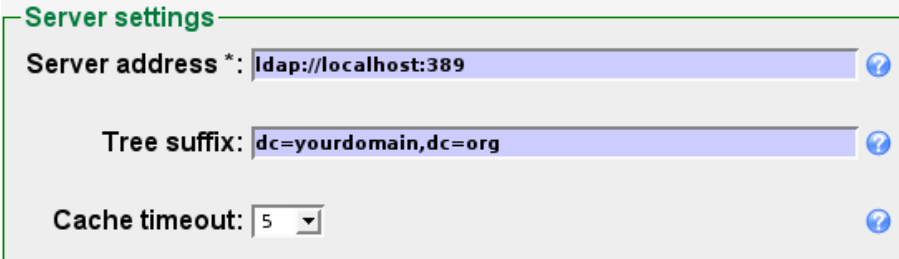


Abbildung 3.6: LAM – Servereinstellungen

Account types and modules – Abbildung 3.7 Hier stellt man die zu verwendenden Kontotypen und Module ein. In **Edit account types** wird festgelegt, welche Art von LDAP-Einträgen verwaltet werden. In einer reinen Linuxumgebung reichen für den Anfang Benutzer- und Gruppenaccounts. Die nicht benötigten Typen sollte man entfernen, da sie sonst ungenutzt in der Übersicht stehen. Die LDAP-Suffixe müssen in jedem der Bereiche entsprechend des eigenen Netzes eingetragen werden (z.B. `dc=localnet`). Neben den schon vordefinierten LDAP-Attributen können eigene hinzugefügt werden.

Unter **Edit modules** werden die Module ausgewählt, welche jeder Objektklasse zur Verfügung stehen sollen. Damit man ein minimales funktionierendes System hat (nur Linux), wählt man im Bereich Benutzer die Module *inetOrgPerson*, *posixAccount* und *shadowAccount* und bei der Gruppe das Modul *posixGroup*.

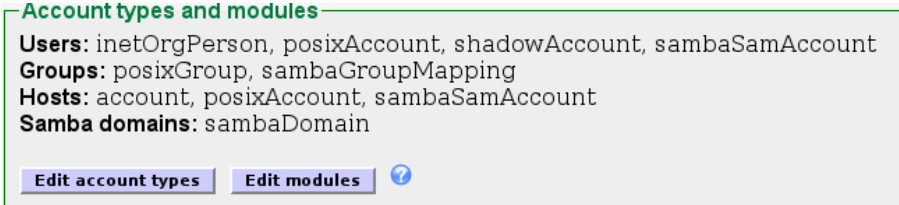


Abbildung 3.7: LAM – Account-Typen

Weitere Accounttypen und Module werden später im Verlauf der erweiterten Konfiguration hinzugefügt und erklärt.

UID- and GID-Bereiche for Unix accounts – Abbildung 3.8 und 3.9 Hier wird der Bereich der zu vergebenden Benutzer-IDs eingestellt. Die Werte sollten sich von denen der Host-IDs unterscheiden, wenn Host-Verwaltung verwendet wird. Ebenso kann die gewünschte Passwort-verschlüsselung gewählt werden. MD5 ist dabei eine gute Wahl.

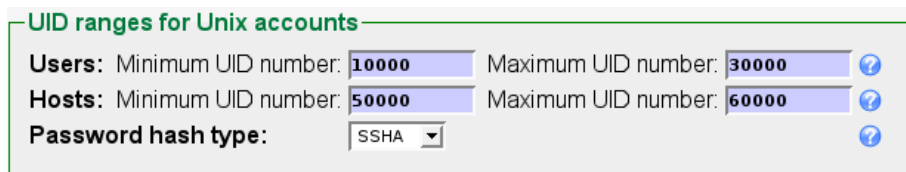


Abbildung 3.8: LAM – Bereich der UIDs

Gruppen-IDs und Benutzer-IDs sollten die gleichen Bereiche sein.

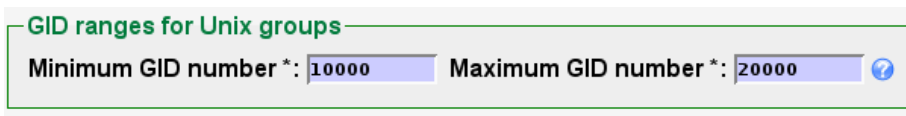


Abbildung 3.9: LAM – Bereich der GIDs

Die voreingestellten Werte sind meist eine gute Wahl und können so bleiben.

List and Language settings – Abbildung 3.10 und 3.11 Mit diesem Wert wird festgelegt, wieviele Einträge (Benutzer, Gruppen, Hosts, ...) in der Übersicht des LAM angezeigt werden.

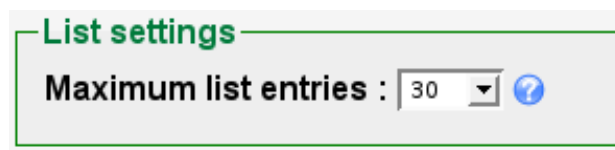


Abbildung 3.10: LAM – Anzahl der anzuzeigenden Einträge

Auswahl der Sprache, mit welcher der LAM bedient wird. Die Einstellung wird nach dem Speichern gültig.




Abbildung 3.11: LAM – Spracheinstellung

Script settings – Abbildung 3.12 An dieser Stelle werden zum im Moment noch keine Einträge vorgenommen. Dies erfolgt mit der Einrichtung des *lamdaemon* (siehe Kapitel 4 ab Seite 15).

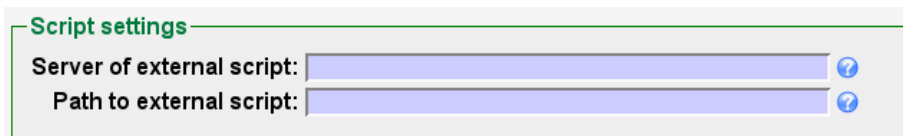


Abbildung 3.12: LAM – Einstellungen für den lamdaemon

Security settings – Abbildung 3.13 Hier werden die Benutzer angegeben, welche den LDAP-Server administrieren dürfen. Sind es mehrere, so müssen diese durch ein ; getrennt sein.

`cn=admin,dc=yourdomain,dc=org` ist der bei der Installation des LDAP-Servers angelegte Administrator. Dies sollte man hier vorerst eintragen, um die weiteren Schritte ausführen zu können. Für `dc=yourdomain,dc=org` sind die eigenen Bezeichnungen einzutragen.

Weiterhin kann hier das Passwort für das aktuelle Profil geändert werden.

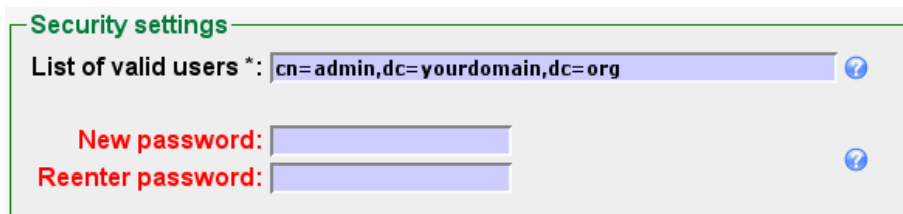


Abbildung 3.13: LAM – Liste der erlaubten Benutzer

Damit sind die Ersteinstellungen auf dieser Seite abgeschlossen und es wird mit **OK** bestätigt. Danach geht es zurück zum Login, welcher nun in der gewünschten Sprache erscheinen sollte.

3.1.2.2 Erster Login

Nachdem die Konfiguration des LDAP Account Manager abgeschlossen ist, kann man sich auf der Login-Seite mit dem Administratorpasswort des LDAP-Servers, welcher bereits dafür konfiguriert sein muss, einloggen. Man landet dann bei der Benutzerübersicht, welche bei einer Neuinstallation des LDAP-Servers leer ist (siehe Abbildung 3.14). Eventuell wird man gefragt, ob die bei der LAM-Profilkonfiguration eingestellten User- und Group-Suffixe, nun angelegt werden sollen. Diese Fragen sind entsprechend zu bestätigen. Verwendet man den LAM für einen bestehenden LDAP-Server, so sollten die Einträge im LDAP-Verzeichnis hier zu sehen sein.

Damit ist die Installation und Konfiguration des LDAP Account Manager abgeschlossen.



Abbildung 3.14: LAM – Nach erstem Login

3.2 Benutzer und Gruppen anlegen

Wird noch geschrieben.

4 lamdaemon

Das Script lamdaemon wird benötigt, um auf dem lokalen oder einem entfernten Rechner Quotas zu managen und/oder Home-Verzeichnisse anzulegen oder zu löschen. Die Konfiguration des lamdaemon muss nicht durchgeführt werden, möchte man die genannten Dinge von Hand erledigen.

4.1 Konfiguration

Damit der LDAP Account Manager auf den lamdaemon zugreifen kann, muss in dem entsprechend Administrationsprofil (siehe auch Abschnitt 3.1.2.1 im Bereich *Script settings* auf Seite 13) der Server und Pfad zum lamdaemon eingetragen werden.

In der ersten Zeile trägt man den DNS-Namen, oder die IP, des Servers ein, auf welchem der lamdaemon ausgeführt werden soll. Die zweite Zeile ist für den vollen Pfad zum lamdaemon. Liegen die Homeverzeichnisse lokal, so kann als IP *172.0.0.1* genommen werden und als Verzeichnis */usr/share/ldap-account-manager/lib/lamdaemon.pl*.

Wie in Abbildung 4.1 zu sehen ist, befindet sich der lamdaemon, und damit die Homeverzeichnisse, auf einem anderen Rechner. Hier wurde die IP des Rechners eingetragen, die Angabe von DNS-Namen ist auch möglich. Der Pfad muss komplett eingetragen werden.

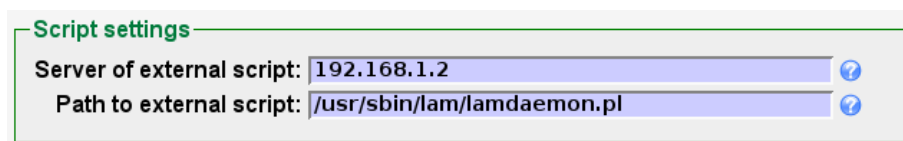


Abbildung 4.1: Einstellungen zur Nutzung des lamdaemon

Der Administrator, welcher sich am LAM einloggt, muss im LDAP-Verzeichnis ein regulärer Unix-Account sein und die Objektklasse *“posixAccount”* und das Attribut *“uid”* enthalten. Der LDAP-Admin, der beim installieren von OpenLDAP angelegt wurde (siehe Abschnitt 6 auf Seite 23), funktioniert nicht an dieser Stelle. Es muss ein Benutzer angelegt werden, welcher am besten den gleichen Namen und auch das Passwort des LDAP-Administrators hat. Im Beispiel wird der Benutzer *admin* mit der Gruppe *lamdaemon* angelegt. Zum Anlegen von Benutzern bitte in Abschnitt 3.2 auf Seite 14 nachlesen. Wichtig ist, die User- und Group-ID für diesen Benutzer unter 1000 zu wählen. Dazu ist es erforderlich, den Bereich der zu verwendenden User-

und Group-IDs (siehe Abschnitt 3.1.2.1 im Bereich *UID- and GID-Bereiche for Unix accounts* auf Seite 12) entsprechend anzupassen.

Bei Debian werden alle Accounts unter UID 1000 nicht für normale Benutzer genommen und erscheinen daher nicht bei Loginmanagern. Im Beispiel wird die UID und GID 900 für *admin:lamdaemon* verwendet. Die weiteren Schritte sollte man erst nach Anlegen dieses Nutzers vornehmen.

4.1.1 Administrator anlegen

Um die Benutzbarkeit des LAM und lamdaemon zu erleichtern, ist es ratsam den lamdaemon-Benutzer zu einem LDAP-Administrator zu machen. Dazu wird der Benutzer im Administrationsprofil (siehe Abschnitt 3.1.2.1 im Bereich *Security settings* auf Seite 13) eingetragen. Im Beispiel (siehe Abbildung 4.2) lautet der Name ebenfalls *admin*.

Abbildung 4.2: Benutzer des lamdaemon

4.1.1.1 LDAP-Anpassung

Damit der neue Benutzer Einträge im LDAP-Baum vornehmen kann, muss er als Administrator eingetragen sein. Daher ist die LDAP-Konfigurationsdatei */etc/ldap/slapd.conf* um folgende (fettgedruckten) Einträge zu ergänzen:

```
/etc/ldap/slapd.conf

# root der LDAP Datenbank
rootdn "uid=admin,dc=organisation"
#rootpw geheim
rootpw {SSHA}oX+4XxtQeiY15CqiE4WQ21XkxNr1

access to attrs=userPassword
    by dn="cn=admin,dc=netz" write
    by dn="uid=admin,ou=user,dc=netz" write
    by anonymous auth
    by self write
```

```
by * none

# The admin dn has full write access, everyone else
# can read everything.
access to *
    by dn="cn=admin,dc=netz" write
    by dn="uid=admin,ou=user,dc=netz" write
    by * read
```

4.1.2 Einstellungen auf dem LAM-Server

Auf dem Rechner mit der LAM-Installation werden zwei zusätzliche Pakete (php4-ssh2 und libssh2) benötigt, damit eine SSH-Verbindung aufgebaut werden kann. Diese sind nicht in Debian/Sarge verfügbar, können aber von dieser Webseite herunter geladen werden: <http://apt.bxlug.be/sarge/not-debian/>

Möchte man die Pakete selber bauen, ist diese Anleitung zu empfehlen: http://apt.bxlug.be/sarge/sources/php-ssh2_0.10-2_i386.build

Am einfachsten ist, man legt die Pakete in einem eigenen Paketserver ab und installiert sie mit folgendem Aufruf:

```
root@debian:# apt-get install libssh2 php4-ssh2
```

Steht kein Paketserver zur Verfügung, können die Pakete aus dem Downloadverzeichnis mit `dpkg -i` installiert werden.

4.1.3 Einstellungen auf dem Homeverzeichnis-Server

Aufgrund der benötigten SSH-Pakete (libssh2, ...) auf dem Rechner mit dem LAM, muss auf dem Rechner mit den Homeverzeichnissen ebenfalls SSH ab Version 4.3x installiert sein. Bei Einsatz von Debian/Sarge kann dieses SSH-Paket von www.backports.org bezogen werden. Ab Debian/Etch ist es direkt enthalten. Befinden sich die Home-Verzeichnisse der Benutzer auf dem gleichen Rechner wie der LDAP Account Manager, so sind die folgenden Schritte lokal anzuwenden.

Um auf einem entfernten Rechner das Script zu verwenden, muss es per Hand auf diesen Rechner kopiert werden. Es befindet sich, bei einer Installation mit dem Paketmanagement, im Verzeichnis `/usr/share/ldap-account-manager/lib/`. Mit `scp` wird es auf den Rechner mit den Homeverzeichnissen kopiert, in diesem Fall nach `/usr/sbin/lam/`.

```
root@debian:~# cd /usr/share/ldap-account-manager/lib/
root@debian:~# scp lamdaemon.pl HOMESERVER://usr/sbin/lam/
```

Es wird vorausgesetzt, dass der `sshd` auf dem Zielrechner am Port 22 lauscht. Ist das nicht der Fall, so kann mit `scp -P PORT . . .` der Port angegeben werden. Jedoch muss, um den `lamdaemon` nutzen zu können, der `sshd` auf dem Port 22 lauschen.

Folgende Pakete müssen zusätzlich installiert werden:

```
root@debian:~# apt-get install libquota-perl sudo ldap-utils
```

Der `lamdaemon` muss mit `root`-Rechten laufen, um Quotas festzulegen und die Homeverzeichnisse anzulegen und/oder zu löschen. Aus diesem Grund wird `sudo` benötigt, in welchem der Benutzer `admin` zur Nutzung des `lamdaemon` eingetragen wird. Die LDAP-Utills werden zur Anmeldung von `admin` verwendet, da es sich um einen LDAP-Account handelt (siehe Abschnitt 4.1.1 auf Seite 16). Mit den nachfolgenden Schritten wird versucht, etwas Sicherheit einzubauen.

In die Datei `/etc/sudoers` wird ein weiterer Eintrag hinzugefügt.

```
root@debian:~# visudo
```

```
admin SERVER= NOPASSWD: /usr/sbin/lam/lamdaemon.pl
```

`admin` ist der im Abschnitt 3.2 auf Seite 14 angelegte Benutzer. Bei `SERVER` ist der DNS-Name des Rechners einzutragen, auf dem der `lamdaemon` ausgeführt werden soll.

Nun muss noch `ssh` angepasst werden. Dabei sind folgende Daten hinzuzufügen, b.z.w. anzupassen:

```
/etc/ssh/sshd_config

ListenAddress 192.168.1.2 # IP an der gelauscht werden soll
AllowGroups lamdaemon root
AllowUsers admin root
PasswordAuthentication yes
```

Wichtig ist, dass der `sshd` auf dem Port 22 lauscht, da derzeit nur dieser Port vom LAM unterstützt wird.

Wenn alles fertig eingerichtet ist, kann man den Zugriff für `root` sperren, in dem man diesen, und die Gruppe `root`, aus der Liste der erlaubten Benutzer löscht und zusätzlich folgende Zeile anpasst:

PermitRootLogin **no**

Dies ist aber nur sinnvoll, wenn lokaler Zugriff auf den PC möglich ist.

4.2 Testen des lamdaemon

Seit Version 1.1.1 des LDAP Account Manager ist ein Test des lamdaemon vorhanden. Dieser prüft, ob der Benutzer vorhanden und der Pfad eingetragen wurde. Weiterhin werden die benötigten ssh-Programme abgefragt und ein Login auf dem Server versucht. Aufrufen kann man den Test nach dem Login über den Menüpunkt **Werkzeuge**. Dort wählt man den letzten Punkt **Test** aus, und dann **Lamdaemon-Test**. Klappt alles, sollte die rechte Hälfte der Anzeige grün sein und ein **OK** anzeigen.

Hinweis: Wird PHP4 benutzt, kann es beim Test der Ausführung des lamdaemon auf dem Zielrechner zu einer Fehlermeldung kommen. Dies sollte in einer neueren Version behoben sein.

5 Apache 2

5.1 Installation von Apache 2

Für den LAM ist es ausreichend, Apache 2 mit den Defaulteinstellungen zu installieren. Alle nötigen Konfigurationen, um später auf den LAM zuzugreifen, werden bei der LAM-Installation vorgenommen.

Neben dem Apache werden auch einige PHP-Module benötigt. In Debian/Sarge steht nur das Paket `php4-ldap` zur Verfügung, welches jedoch vorrangig das Paket `apache-common` verlangt. Wird auf dem Server der Apache 1.3 verwendet, stellt dies kein Problem dar. Soll, wie hier gemacht, Apache 2 Verwendung finden, ist eine ausführlichere Angabe der zu installierenden Pakete nötig.

```
root@debian:~# apt-get install apache2 apache2-common  
apache2-mpm-prefork apache2-utils libapache2-mod-php4  
php-fpdf php4 php4-common php4-ldap php4-mhash  
php4-mcrypt
```


6 LDAP

Das Lightweight Directory Access Protocol (LDAP) vermittelt zwischen sog. LDAP-Clients und einem Verzeichnisdienst, welcher verschiedene Informationen (Nutzerverwaltung, Telefonbuch, DNS-Einträge, ...) vorhält und diese auf Anfrage der Clients übermittelt. Die Daten werden in einer Datenbank als Baumstruktur abgelegt.

6.1 Installation und Konfiguration des Servers

OpenLDAP kann unabhängig von anderen Diensten auf einem eigenen Rechner, oder im vorliegenden Fall, auf dem gleichen Rechner mit dem Apache oder den Honeerverzeichnissen, installiert werden. Die Kommunikation erfolgt unverschlüsselt. Auf die Einrichtung einer verschlüsselten Kommunikation wird in Kapitel 7 auf Seite 33 eingegangen. Für den Moment soll aber die Standardvariante ausreichen. Dabei gibt es ausreichend Fehlerpotential als das man sich weitere Fehlermöglichkeiten dazu nehmen muss.

6.1.1 Installation

Die Installation unter Debian kann einfach mit `apt-get` erfolgen. Dabei werden der LDAP-Server `slapd`, Hilfsprogramme (`ldap-utils`) und die Datenbankprogramme installiert.

```
root@debian:~# apt-get install slapd ldap-utils db4.2-util
```

Die während der Installation auftretenden Fragen sind entsprechend des eigenen Netzes zu beantworten.

Bei der Grundinstallation des `slapd` gibt man ein Passwort für den Administrator des LDAP-Servers an. Der zugehörige Benutzername ist `admin`. Im Verzeichnisbaum sieht der Eintrag folgendermaßen aus, angelehnt an die weiter hinten folgende Konfigurationsdatei:

```
cn=admin,dc=organisation
```

Der Administrator wird direkt in der Datenbank gespeichert. Es ist dafür kein weiterer Eintrag in der Konfigurationsdatei (`slapd.conf`) nötig.

6.1.2 Konfiguration

In der Grundkonfiguration läuft der LDAP-Daemon (`slapd`) (meist) als `root`. Dies stellt ein Sicherheitsrisiko dar, daher soll der Dienst auf den Namen eines normalen Nutzers laufen. Die Zugriffe auf den Rechner sollten mittels `inetd` oder `xinetd` eingeschränkt und nur für den freizugebenden Dienst (in diesem Fall der `slapd` und bestimmte Rechner(gruppen) erlaubt sein.

6.1.2.1 Zugriffsrechte auf den `slapd`

Im `inetd` oder `xinetd` per Default der Zugriff für alle Hosts gesperrt sein. Durch folgenden Eintrag für den `inetd` läßt sich der Zugriff auf den `slapd` für das Netz 192.168.1.0/24 freischalten:

```
root@debian:~# echo "slapd: 192.168.1.0/24 127.0.0.1" >>
/etc/hosts.allow
```

Die Adresse ist entsprechend dem eigenen Netz anzupassen.

6.1.2.2 Besitzer und Rechte des `slapd` anpassen

Der `slapd` läuft in der Grundeinstellung mit Benutzer und Gruppe `root`. Daher wird der LDAP-Daemon einem Benutzer und einer Gruppe `ldap` zugeordnet.

Mit den Kommandos `adduser` und `addgroup` werden die entsprechenden Konten angelegt. Zuerst wird die Systemgruppe `ldap` erstellt, welche die Gruppen-ID (gid) 600 erhält.

```
root@debian:~# addgroup system gid 600 ldap
```

Als nächstes wird der Systembenutzer `ldap` erstellt. Dieser bekommt kein Homeverzeichnis, die Benutzer-ID (uid) 600 und wird gleich in die Gruppe `ldap` eingefügt. Das Passwort und der Login wird für diesen Benutzer deaktiviert.

```
root@debian:~# adduser system no-create-home uid 600
ingroup ldap disabled-password disabled-login ldap
```

Nun muss dem `slapd` noch mitgeteilt werden, dass er beim Systemstart mit dem Benutzer und der Gruppe `ldap` starten soll. Dazu sind zwei Einträge in der Datei `/etc/default/slapd` nötig.

```
/etc/default/slapd

...
SLAPD_USER=ldap
SLAPD_GROUP=ldap
...
```

Weiterhin muss beim Eintrag `TRY_BDB_RECOVERY` ein **no** geschrieben werden, da sonst die Datenbank-Dateien wieder dem Benutzer `root:root` mit den Rechten `0600` gegeben werden. Damit hätte der `slapd` keinen Zugriff darauf und würde nicht starten.

Bei `SLAPD_SERVICES` muss neben dem Eintrag für den `localhost` auch die IP des Servers stehen, welche aus dem Netzwerk erreichbar ist. Ein solcher Eintrag könnte so aussehen:

```
/etc/default/slapd

...
SLAPD_SERVICES="ldap://127.0.0.1:389/ ldap://192.168.1.1:389/"
...
```

Es ist auch möglich an dieser Stelle den DNS-Namen des LDAP-Servers einzutragen.

Der Eigentümer und die Rechte mehrerer Dateien müssen nun noch angepasst werden, damit der `slapd` darauf zugreifen kann. Dateien aus denen der `slapd` lesen muss, aber nicht der Rest der Welt, bekommen die Eigentümer `root:ldap`. Muss der `slapd` nicht in die entsprechende Datei schreiben, bekommt die Gruppe nur das Leserecht. Einzig `root` wird weiter Schreibrechte haben.

Zunächst wird die Konfigurationsdatei entsprechend bearbeitet.

```
root@debian:~# chown root:ldap /etc/ldap/slapd.conf
chmod 0640 /etc/ldap/slapd.conf
```

Als nächstes ist das Verzeichnis in `/var` dran.

```
root@debian:~# chown root:ldap /var/run/slapd/
chmod 0770 /var/run/slapd/
```

Hierbei drauf achten, dass in der `slapd.conf` der Pfad zum `argsfile` entsprechend zu `/var/run/slapd/` geändert wird.

Nun muss noch die Datenbank die entsprechenden Rechte bekommen.

```
root@debian:~# chown -R root:ldap /var/lib/ldap
chmod 0770 /var/lib/ldap
```

In diesem Verzeichnis müssen die Rechte der Dateien entsprechend angepasst werden. Dabei sollte die Gruppe die gleichen Rechte bekommen wie der Eigentümer. Mit einem kleinen Bash-Einzeiler, ausgeführt in diesem Verzeichnis (`/var/lib/ldap/`), lässt sich das schnell erledigen.

```
root@debian:~# for i in *; do chmod g+rw $i; done
```

Wurden die Schritte durchgeführt, ist der LDAP-Server soweit vorbereitet, um die weitere Konfigurationen mit dem LAM zu erledigen.

Nachfolgend eine *slapd.conf*-Beispieldatei. Die Originaleinträge sind enthalten.

```

/etc/ldap/slapd.conf

# This is the main slapd configuration file. See
# slapd.conf(5) for more
# info on the configuration options.

# Global Directives:

# Features to permit
#allow bind_v2

# Schema and objectClass definitions
include      /etc/ldap/schema/core.schema
include      /etc/ldap/schema/cosine.schema
include      /etc/ldap/schema/nis.schema
include      /etc/ldap/schema/inetorgperson.schema

# Schema check allows for forcing entries to
# match schemas for their objectClasses's
schemacheck  on

# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile      /var/run/slapd/slapd.pid

# List of arguments that were passed to the server
argsfile     /var/run/slapd/slapd.args

# Read slapd.conf(5) for possible values
loglevel     0

# Where the dynamically loaded modules are stored
modulepath   /usr/lib/ldap
moduleload   back_bdb

# Specific Backend Directives for bdb:
# Backend specific directives apply to this backend until
# another 'backend' directive occurs

```

```
backend          bdb
checkpoint 512 30

# Specific Backend Directives for 'other':
# Backend specific directives apply to this backend until
# another 'backend' directive occurs
#backend

# Specific Directives for database #1, of type bdb:
# Database specific directives apply to this database until
# another 'database' directive occurs
database          bdb

# The base of your directory in database #1
suffix            "dc=organisation"

# Where the database file are physically stored
# for database #1
directory         "/var/lib/ldap"

# Indexing options for database #1
index             objectClass eq

# Save the time that the entry gets modified,
# for database #1
lastmod           on

# Where to store the replica logs for database #1
# relogfile       /var/lib/ldap/replug

# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
# admin entry below
# These access lines apply to database #1 only
access to attrs=userPassword
    by dn="cn=admin,dc=organisation" write
    by anonymous auth
    by self write
    by * none
```

```
# Ensure read access to the base for things like
# supportedSASLMechanisms. Without this you may
# have problems with SASL not knowing what
# mechanisms are available and the like.
# Note that this is covered by the 'access to *'
# ACL below too but if you change that as people
# are wont to do you'll still need this if you
# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read

# The admin dn has full write access, everyone else
# can read everything.
access to *
    by dn="cn=admin,dc=organisation" write
    by * read

# For Netscape Roaming support, each user gets a roaming
# profile for which they have write access to
#access to dn=".*,ou=Roaming,o=morsnet"
#    by dn="cn=admin,dc=organisation" write
#    by dnattr=owner write

# Specific Directives for database #2, of type 'other'
# (can be bdb too):
# Database specific directives apply to this databasse until
# another 'database' directive occurs
#database

# The base of your directory for database #2
#suffix          "dc=debian,dc=org"
```

6.2 Installation und Konfiguration des Clients

Damit LDAP-Clients auf den Verzeichnisbaum zugreifen können, müssen einige Programmpakete installiert und Konfigurationsdateien angepasst werden.

6.2.1 Installation

Die benötigten Pakete auf dem Client installiert man am besten mit `apt-get`:

```
root@debian:~# apt-get install libldap2 ldap-utils
```

6.2.2 Konfiguration

Im Verzeichnis `/etc/ldap` liegt die Client-Konfigurationsdatei `ldap.conf`. Diese bekommt folgende Einträge:

```
/etc/ldap/ldap.conf

BASE    dc=organisation
URI     ldap://server
```

Mit `BASE` wird der oberste Eintrag des LDAP-Verzeichnisbaumes angegeben. Dies ist der Eintrag, der in der `/etc/ldap/slapd.conf` auf dem LDAP-Server hinter `suffix` angegeben wurde. Hinter `URI` wird die Adresse des LDAP-Servers eingetragen. Durch die Angabe des Protokolls `ldap://` ist es nicht nötig einen Port anzugeben. Der Standardport ist 389.

Damit die Authentifizierung am Client mittels `PAM` erfolgen kann, sind noch folgende Pakete zu installieren:

```
root@debian:~# apt-get install libnss-ldap libpam-ldap
```

Bei der Installation werden einige Fragen gestellt, die wie folgt zu beantworten sind:

LDAP Server host. 127.0.0.1 (oder die IP-Adresse des LDAP-Servers)

The distinguished name of the search base. `dc=organisation` (wie in der `/etc/ldap/slapd.conf` auf dem LDAP-Server unter `suffix` angegeben)

LDAP version to use. 3 (aus Sicherheitsgründen sollte Version 3 verwendet werden)

database requires login No

make configuration readable/writeable by owner only No (werden Passwörter in der Datei `/etc/libnss-ldap.conf` verwendet, was bei dieser Installation nicht vorgesehen ist, so muss hier mit `Yes` geantwortet werden. Ein lokaler `root` hat dann Zugriff auf diese Datei, was bei unterschiedlichen Administratoren sicher nicht erwünscht ist.)

Make local root Database admin. No (es wird keine lokale root-Datenbank angelegt, da dies ein Sicherheitsrisiko darstellt und für diese Installation nicht benötigt wird)

Database requires logging in. No

Local crypt to use when changing passwords. md5

Als nächster Schritt folgt die Anpassung weiterer Konfigurationsdateien, damit der Client auf den LDAP-Server zugreift.

```
/etc/nsswitch.conf

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch
# functionality. If you have the 'glibc-doc' and
# 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information
# about this file.

passwd:          files ldap
group:           files ldap
shadow:         files ldap

hosts:          files dns
networks:       files

protocols:     db files
services:      db files
ethers:        db files
rpc:           db files

netgroup:      nis
```

Ein kleiner Funktionstest an dieser Stelle sollte keinen Fehler erzeugen:

```
root@linux:~# getent passwd <BENUTZER>
```

Es sollte nun eine Zeile wie diese ausgegeben werden:

```
BENUTZER:x:10002:10002:NAME:/home/BENUTZER:/bin/bash
```

Lässt man den Benutzernamen weg, werden der Inhalt der Datei */etc/passwd* und die Einträge im LDAP ausgegeben.

6.2.3 Konfigurieren von PAM

Damit man sich mittels der Accounts im LDAP an einem Client anmelden kann, sind noch einige Handgriffe in den Konfigurationsdateien von PAM nötig. Im Verzeichnis */etc/pam.d/* liegen die Dateien der verschiedenen Dienste, welche den Zugriff und anderes regeln. Sind diese Dateien vorhanden, wird */etc/pam.conf* ignoriert und kann leer bleiben.

Sind die Dateien in */etc/pam.d/* unverändert, so müssen lediglich die folgenden 4 Dateien mit diesen Einträgen versehen werden:

```
/etc/pam.d/common-account

#
# /etc/pam.d/common-account - authorization settings common
# to all services
#
account sufficient pam_ldap.so
account required pam_unix.so
```

```
/etc/pam.d/common-auth

#
# /etc/pam.d/common-auth - authentication settings common
# to all services
#
auth sufficient pam_ldap.so
auth required pam_unix.so nullok_secure use_first_pass
```

```
/etc/pam.d/common-password

#
# /etc/pam.d/common-password - password-related modules common
# to all services
#
# This file is included from other service-specific PAM
# config files, and should contain a list of modules that
# define the services to be used to change user passwords.
# The default is pam_unix
#
```

```
password      sufficient      pam_ldap.so
password     required pam_unix.so md5 use_first_pass
```

```
/etc/pam.d/common-session

#
# /etc/pam.d/common-session - session-related modules common
# to all services
#
session sufficient      pam_ldap.so
session required       pam_unix.so
```

Ein anschließender Login auf der Konsole oder im Anmeldemanager der Desktopumgebung (z.B. kdm) sollte erfolgreich sein.

7 Verschlüsselung mit SSL/TLS

Die Kommunikation mit dem LDAP Account Manager und auch mit dem LDAP-Server sollte stets verschlüsselt erfolgen. Zu diesem Zweck wird eine SSL-Verschlüsselung eingerichtet.

In der vorliegenden Konfiguration befinden sich der Apache-Web- und LDAP-Sever auf unterschiedlichen Rechnern. Daher muss man für beide Dienste jeweils ein SSL-Zertifikat erstellen. Steht nur ein Rechner für beides zur Verfügung, kann das gleiche Zertifikat verwendet werden.

Auf den Servern muss das Open-SSL-Paket¹ installiert sein. Dieses geschieht auf allen Rechnern, auf denen dies benötigt wird, mit folgendem Aufruf:

```
root@linux:~# apt-get install openssl
```

Um SSL nutzen zu können, muss ein Zertifikat erstellt werden. Im Internet sind eine Menge Howtos verfügbar, welche die Erstellung eines Zertifikates auf unterschiedliche Weise beschreiben. Das OpenSSL-Paket liefert Scripte mit, durch die sich ein Zertifikat leicht erstellen lässt. Dabei kann man die Standardeinstellungen übernehmen, oder die entsprechenden Bereiche in den Scripten durch eigene benötigte Werte ersetzen. Die Vorgehensweise ist dabei von Distribution zu Distribution unterschiedlich. Hier wird beschrieben, wie das Zertifikat auf der Kommandozeile mit OpenSSL erstellt wird. Es folgt anschließend eine Beschreibung, um das Zertifikat in die Dienste einzubinden.

Weiterhin gibt es grafische Frontends zum erstellen von Zertifikaten. Eines davon ist *tinyca*.

Hinweis: Die Erstellung des Zertifikates für den Apache wird mit dem bei Apache mitgelieferten Tool gemacht. Es folgt in einer späteren Version dieses Handbuches, wie man das mit Konsolenbefehlen erstellte Zertifikat einbindet.

Bevor mit der Erstellung des Zertifikates begonnen wird, sollte eine Datei mit Zufallszeichen und einer Mindestgröße von 1000 Byte erstellt werden. Diese wird für den privaten Schlüssel benötigt. Mit folgendem Befehl lässt sich die Datei erzeugen, der Vorgang kann mit Ctrl+C beendet werden:

```
root@linux:~# cat /dev/random > /tmp/zufall
```

Alternativ kann man die Schlüsselerstellung durch Systemaktivitäten unterstützen, indem man z.B. ein Image aus */dev/zero* erzeugt. Den Befehl startet man, wenn der key erzeugt wird.

```
root@linux:~# dd if=/dev/zero of=/tmp/x bs=1024 count=100000
```

¹<http://www.openssl.org>

7.1 CA Zertifikat und Schlüssel

Im *root*-Verzeichnis sollte ein Ordner erstellt werden, in welchem die benötigten privaten Schlüssel und Zertifikate erstellt und gespeichert werden. Zusätzlich wird gleich die Zertifikatsverzeichnisstruktur angelegt, wie sie in der Konfigurationsdatei von `openssl` definiert ist.

```
root@linux:~# mkdir -p /root/cert/demoCA/newcerts
```

Nun wird der Ordner **cert** nur für *root* les- und schreibbar gemacht, damit die erstellten Schlüssel und Zertifikate relativ sicher sind.

```
root@linux:~# chmod 0700 /root/cert
```

Weiterhin werden noch zwei Dateien angelegt:

```
root@linux:~# touch /root/cert/demoCA/index.txt
root@linux:~# echo "01" > /root/cert/demoCA/serial
```

In der Datei **index.txt** werden die signierten Zertifikate eingetragen und **serial** enthält eine Seriennummer. Nach jeder Signierung wird diese um eins erhöht.

Im Verzeichnis **/root/cert/** wird nun eine Root-CA (Certificate Authority) mit privatem Schlüssel erstellt. Diese wird später zum signieren des Server-Zertifikates verwendet.

```
root@linux:~# openssl req -x509 -days 8000 -newkey
rsa:8192 -keyout RootcaKey.pem -out RootcaCert.pem
```

Die verwendeten Optionen haben folgende Bedeutung:

req Das `req` Kommando erzeugt und bearbeitet zuerst eine Zertifikat Anfrage im PKCS#10 Format. Es können zusätzlich auch eigene selbst signierte Zertifikate erzeugen für die Benutzung als root CAs zum Beispiel.

-x509 Wenn die Option `-x509` benutzt wird, wird hierdurch die Anzahl der Tage definiert für die das Zertifikate zertifiziert ist. Der Standardwert ist 30 Tage.

-days Anzahl der Tage, wie lange das Zertifikat gültig ist. Default sind 30 Tage.

-newkey Diese Option erzeugt eine neue Zertifikatsanfrage und einen neuen privaten Schlüssel (private key). Das Argument kann zwei Formen haben: **rsa: nbits**, hierbei ist **nbits** die Anzahl der Bits, welche einen RSA Key mit der Größe **nbits** erzeugt. **dsa: filename**, erzeugt einen DSA Schlüssel und benutzt die Parameter in der Datei **filename**.

- keyout** Dies gibt den Dateinamen an, in welchen der neue private Schlüssel geschrieben wird. Ist diese Option nicht gesetzt, wird der Dateiname aus der Konfigurationsdatei von *openssl* genommen.
- out** Diese spezifiziert den Ausgabedateinamen in welche geschrieben werden soll, oder die Standardausgabe.

Man wird nun nach einem Passwort für den privaten Schlüssel gefragt. Dieses sollte ausreichend lang und kompliziert sein. Anschließend werden Informationen zur Organisation erfragt und das Zertifikat selber signiert.

7.2 Server-Zertifikat und Schlüssel

Für das eigentliche Serverzertifikat muss nun ein privater Schlüssel und eine Zertifikatsanforderung erstellt werden. Der Schlüssel wird ohne Passwort gespeichert, da dieses sonst beim Systemstart eingegeben werden müsste, was bei einem Server eher unpraktisch wäre. Die Zertifizierungsanfrage (Certificate Signing Request) wird anschließend mit der erstellten Root-CA zertifiziert. Daraus entsteht das eigentliche Serverzertifikat.

```
root@linux:~# openssl req -nodes -newkey rsa:8192  
-keyout serverKey.pem -out serverReq.pem
```

Die verwendeten Optionen haben folgende Bedeutung:

- req** Das req Kommando erzeugt und bearbeitet zuerst eine Zertifikat Anfrage im PKCS#10 Format. Es können zusätzlich auch eigene selbst signierte Zertifikate erzeugen für die Benutzung als root CAs zum Beispiel.
- nodes** Wenn diese Option spezifiziert ist, dann wird der private Schlüssel der erzeugt wird, nicht verschlüsselt.
- newkey** Diese Option erzeugt eine neue Zertifikatsanfrage und einen neuen privaten Schlüssel (private key). Das Argument kann zwei Formen haben: **rsa: nbits**, hierbei ist **nbits** die Anzahl der Bits, welche einen RSA Key mit der Größe **nbits** erzeugt. **dsa: filename**, erzeugt einen DSA Schlüssel und benutzt die Parameter in der Datei **filename**.
- keyout** Dies gibt den Dateinamen an, in welchen der neue private Schlüssel geschrieben wird. Ist diese Option nicht gesetzt, wird der Dateiname aus der Konfigurationsdatei von *openssl* genommen.
- out** Diese spezifiziert den Ausgabedateinamen in welche geschrieben werden soll, oder die Standardausgabe.

Es werden wieder Informationen zur Organisation abgefragt. Wichtig ist, dass bei der Frage nach dem Common Name der volle DNS-Name des Rechners/Dienstes angegeben wird, für welchen das Zertifikat ausgestellt werden soll. Beispiel: *mein.server.net*.

7.2.1 Zertifikat signieren

Mit folgendem Befehl beginnt die Signierung durch die selbsterstellte CA:

```
root@linux:# openssl ca -in serverReq.pem -days 8000
-out serverCert.pem -notext -cert RootcaCert.pem
-keyfile RootcaKey.pem
```

Die verwendeten Optionen haben folgende Bedeutung:

- ca** Das Kommando **ca** ist eine kleinst mögliche CA-Anwendung. Sie signiert Zertifizierungsanfragen in verschiedenster Form und generiert CRLs. Desweiteren enthält es eine Text-Datenbank ausgestellter Zertifikate und deren Status.
- in** Die Angabe der Datei, welche eine Zertifizierungsanfrage enthält, die durch die CA signiert werden soll.
- days** Anzahl der Tage, wie lange das Zertifikat gültig ist.
- out** Gibt den Dateinamen an, in den das Zertifikat geschrieben wird. Die Default-Ausgabe ist die Standardausgabe. Die Details des Zertifikates werden in die Datei geschrieben (oder auf die Ausgabe).
- notext** Keine Ausgabe der Textausgabe des Zertifikates in die Ausgabedatei.
- cert** Die CA-Zertifikatsdatei, mit welcher die Anfrage signiert werden soll.
- keyfile** Der private Schlüssel der CA-Datei, mit dem die Anfrage signiert werden soll.

Es werden alle eingetragenen Daten angezeigt und man muss diese bestätigen. Im Verzeichnis liegt nun die Datei *serverCert.pem*, welche das Serverzertifikat ist.

7.3 Zertifikat einrichten

Um das Zertifikat verwenden zu können, müssen die Dateien nach */etc/ssl/* kopiert werden. Nach */etc/ssl/certs/* werden *RootcaCert.pem* und *serverCert.pem* kopiert.

```
root@linux:# cp RootcaCert.pem serverCert.pem /etc/ssl/certs
```

Der private Schlüssel des Serverzertifikates (*serverKey.pem*), wird nach */etc/ssl/private* kopiert. Die Rechte müssen so angepasst werden, dass nur *root* und der Benutzer, welcher für den entsprechenden Prozess, der SSL nutzen soll, eingerichtet wurde, darauf zugreifen dürfen. Die Rechteanpassung für den jeweiligen Dienst wird im entsprechenden Abschnitt erklärt. Der Schlüssel ist nicht durch ein Passwort geschützt und darf nicht in unbefugte Hände gelangen. Der Schlüssel wird vorerst nur für *root* lesbar sein.

```
root@linux:# cp serverKey.pem /etc/ssl/private
root@linux:# chmod 0400 /etc/ssl/private/serverKey.pem
```

Damit ist die Erstellung eines Zertifikates abgeschlossen und es kann in die jeweiligen Dienste eingebunden werden. Zur Sicherheit sollte das `/root/cert/`, in welchem die privaten Schlüssel der CA und des Serverzertifikates und auch die CA und Zertifikate liegen, nur für `root` les- und beschreibbar sein.

```
root@linux:# chmod 0700 /root/cert/
```

7.4 SSL und Apache 2

Um SSL für den Apachen einzurichten und zu aktivieren, werden im Paket `apache2-common` einige Scripte mitgeliefert, welche die Arbeit erleichtern.

Mit dem Kommando

```
root@linux:# a2enmod ssl
```

wird die SSL-Unterstützung im Apache aktiviert.

Der Aufruf

```
root@linux:# apache2-ssl-certificate
```

startet die Erstellung des Zertifikates. Dabei müssen einige Fragen beantwortet werden.

Bei der Frage nach dem Servernamen ist es wichtig, den vollen DNS-Namen des Servers anzugeben.

Mit dem Zusatz `days TAGE` kann man das Zertifikat für einen bestimmten Zeitraum erstellen.

Bisher lauscht der Apache-Webserver auf Port 80. Nun soll zusätzlich auf dem Port 443 gelauscht werden. Dazu ist ein Eintrag in der Datei `ports.conf` nötig.

```
root@linux:# echo "Listen 443" >> /etc/apache2/ports.conf
```

Damit der Apache auf eine Anfrage auf dem Port 443 reagiert, muss eine entsprechende Konfigurationsdatei vorhanden sein. Um diese zu erstellen wird zuerst eine Default-Datei kopiert.

```
root@linux:# cp /etc/apache2/sites-available/default
/et/apache2/sites-available/ssl-enable
```

Die Datei muss entsprechend der Umgebung angepasst werden. Wichtig sind die SSL-Einträge am Ende der Konfigurationsdatei.

```
/etc/apache2/sites-available/ssl-enable

<VirtualHost 192.168.1.208:443>
    ServerName diamant.homenet
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
        # This directive allows us to have apache2's
# default start page in /apache2-default/,
# but still have / go to the right place
        RedirectMatch ^/$ /apache2-default/
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice,
    # warn, error, crit, alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined
    ServerSignature On

Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
```

```
Options Indexes MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.pem
SSLProtocol all
SSLCipherSuite HIGH:MEDIUM
</VirtualHost>
```

Anschließend erfolgt die Aktivierung der Seite. Dabei wird mit dem folgenden Aufruf ein symbolischer Link in `/etc/apache2/sites-enabled/` erstellt.

```
root@linux:~# a2ensite ssl-enable
```

Nach dem Neustart des Apachen (oder einlesen der Konfigurationsdateien), sollte der Verbindungsaufbau über SSL möglich sein.

```
root@linux:~# /etc/init.d/apache2 restart
```

7.5 SSL und LDAP

Der LDAP-Server lauscht nach der Neuinstallation in der Standardeinstellung auf dem Port 389. Die Verbindung findet unverschlüsselt statt. Somit kann ein potentieller Angreifer den Datenverkehr mitschneiden. Um dem vorzubeugen, soll die LDAP-Verbindung mittels Open-SSL abgesichert werden.

Wie in Abschnitt 7.1 und 7.2 beschrieben, sollten die benötigten Zertifikate erstellt und an die entsprechende Stelle im SSL-Verzeichnis kopiert sein. In der LDAP-Konfigurationsdatei müssen folgende Einträge gemacht werden, um LDAP durch eine SSL-Verschlüsselung abzusichern:

```
/etc/ldap/slapd.conf

...
#Specify ciphers
```

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2:+SSLv3
#TLS keyfile locations
TLSCACertificateFile /etc/ssl/certs/RootcaCert.pem
TLSCertificateFile /etc/ssl/certs/serverCert.pem
TLSCertificateKeyFile /etc/ssl/private/serverKey.pem
...
```

Hat man für den LDAP-Server-Prozess (`slapd`), wie in Abschnitt 6.1.2.2 auf Seite 24 beschrieben, einen neuen Systembenutzer angelegt und lässt den Dienst nicht als `root` laufen, so muss der private Schlüssel des Serverzertifikates für den Benutzer `ldap` lesbar sein. Um dies zu erreichen, ist es notwendig, die Gruppe des Schlüssels auf die des Benutzers `ldap` zu ändern. Diese Gruppe wurden ebenfalls `ldap` genannt.

```
root@linux:# chown root:ldap /etc/ssl/private/server.key
root@linux:# chmod 0440 /etc/ssl/private/server.key
```

In der `/etc/default/slapd` wird der SSL-Port eingetragen, damit LDAP auch an diesem lauscht.

```
SLAPD_SERVICES="ldap://127.0.0.1:389/
ldap://192.168.1.1:389/ ldaps://192.168.1.1:636/"
```

Nach dem Neustart des `slapd` lauscht der LDAP-Server nun auch auf Port 636. Um zu testen ob das Zertifikat erkannt wird, kann man auf dem Server oder einem Clientrechner folgendes Kommando ausführen:

```
root@linux:# openssl s_client -connect server:636
```

Es sollten nun einige Ausgaben zum Zertifikat und öffentlichen Schlüssel erscheinen.

7.5.1 SSL auf dem LDAP-Client

Wird noch geschrieben.