



# LDAP-Server zur Benutzerverwaltung und -authentifizierung unter Linux und Windows

Stefan Knobel (Matrikelnummer 237368)

Daniel Wendler (Matrikelnummer 837576)

**Sonntag, 31. Oktober 2004**

1	Allgemeine Erläuterungen .....	5
1.1	Was ist OpenLDAP? .....	5
1.2	Wofür braucht man OpenLDAP? .....	5
1.3	Unterschied zwischen Verzeichnisdienst und konventioneller Datenbank ...	6
1.4	Wie kann LDAP die Arbeit in einem Netzwerk erleichtern? .....	7
1.5	Verwendete Software .....	7
1.5.1	Berkeley DB .....	7
1.5.2	OpenLDAP .....	7
1.5.3	OpenSSL .....	7
1.5.4	NSS_LDAP .....	8
1.5.5	PAM_LDAP .....	8
1.5.6	SAMBA .....	9
1.5.7	SMBLDAP-Tools .....	9
2	Installation der benötigten Software .....	10
2.1	Berkeley Datenbank .....	10
2.2	OpenSSL .....	11
2.3	OpenLDAP .....	11
2.4	NSS_LDAP .....	12
2.5	PAM .....	13
2.6	Samba .....	13
2.7	Installation erforderlicher Perlmodule .....	14
2.8	SambaLDAP-Tools .....	16
2.9	JDK .....	16
2.10	LDAP-Browser .....	17
3	Konfiguration der Komponenten .....	18
3.1	Berkeley DB .....	18
3.2	OpenSSL .....	18
3.3	OpenLDAP .....	18
3.4	NSS .....	20
3.5	Populieren der Datenbank .....	20
3.6	Testen der bisherigen Konfiguration .....	21
3.7	NFS-Server konfigurieren .....	21
3.8	NFS-Clients konfigurieren .....	21
3.9	Samba konfigurieren .....	22

3.10	SambaLDAP-Tools konfigurieren.....	26
3.11	Testen der Konfiguration, Hinzufügen Windows-Rechners in die Domäne	28
4	Das System mit SSL absichern.....	29
4.1	CA (Certificate Authority).....	29
4.2	Erstellen des Root-Zertifikates (CA).....	29
4.2.1	Vorbereitungen für OPENSSL.....	29
4.2.2	Erstellen des Schlüssels der CA.....	29
4.2.3	Signieren der CA.....	29
4.2.4	Erstellen des LDAP-Server-Schlüssels.....	29
4.2.5	Erstellen des LDAP-Server-Zertifikates.....	30
4.2.6	Signieren des LDAP-Server-Zertifikates mit der CA.....	30
4.2.7	Erstellen des Schlüssels für SMBLDAP-Tools.....	30
4.2.8	Erstellen des Zertifikates für SMBLDAP-Tools.....	30
4.2.9	Signieren des SMBLDAP-Tools-Server-Zertifikates mit der CA.....	30
4.3	Erstellen der Zertifikate für LDAP-Clients.....	30
4.4	Anpassen der Konfigurationsdatei "slapd.conf".....	31
4.5	Anpassen der Konfigurationsdatei „smbldap-tools.conf“.....	31
4.6	Anpassen der Konfigurationsdatei "ldap.conf", (Client).....	32
4.7	Anpassen der Konfigurationsdatei "smb.conf".....	32
5	Appendix.....	33
5.1	OpenLDAP als Adressbuch für Outlook bzw. andere eMail-Clients.....	33
5.2	OpenLDAP.....	34
5.2.1	/etc/rc5.d/S49openldap.....	34
5.2.2	/etc/rc5.d/K49openldap.....	34
5.2.3	/etc/openldap/slapd.conf.....	34
5.2.4	/etc/ldap.conf (Client).....	36
5.3	/etc/nsswitch.....	36
5.4	/etc/pam.d/login.....	37
5.5	SAMBA.....	37
5.5.1	/etc/samba/smb.conf.....	37
5.6	SSL.....	39
5.6.1	/secure/ca.conf.....	39
5.7	SambaLdapTools.....	40
5.7.1	/etc/smbldap-tools/smbldap.conf.....	40
5.7.2	/etc/smbldap-tools/smbldap_bind.conf.....	42

6	Fazit .....	43
7	Quellen.....	44

## 1 Allgemeine Erläuterungen

### 1.1 Was ist OpenLDAP?

Beim **Lightweight Directory Access Protocol** handelt es sich um ein Protokoll, welches die Abfrage von Informationen eines Verzeichnisdienstes<sup>1</sup> ermöglicht.

**LDAP** wurde mit dem Ziel entwickelt, Verzeichnisdienste einfacher und somit populärer zu gestalten. LDAP entstand ursprünglich aus dem Standard **X.500** oder auch **DAP** genannt. Das ursprüngliche X.500 war sehr lange die Referenz für Verzeichnisdienste, wurde jedoch von LDAP verdrängt, da es zu komplex war und nicht mit TCP/IP zusammenzuarbeiten konnte. Trotzdem ist LDAP in der Lage auf X.500 Verzeichnisse zuzugreifen, jedoch nicht in vollem Funktionsumfang der von X.500 vorgegeben wurde.

LDAP wurde in den frühen 90ern an der University of Michigan implementiert. Es gibt verschiedene kommerzielle LDAP-Server aber auch freie Versionen wie das hier verwendete OpenLDAP.



Kommunikation über TCP/IP

### 1.2 Wofür braucht man OpenLDAP?

Heutzutage müssen Anwendungen oftmals nicht mehr nur auf lokale Ressourcen bzw. Ressourcen im LAN zugreifen, sondern auf Ressourcen irgendwo im Internet bzw. im großen Firmennetzwerk.

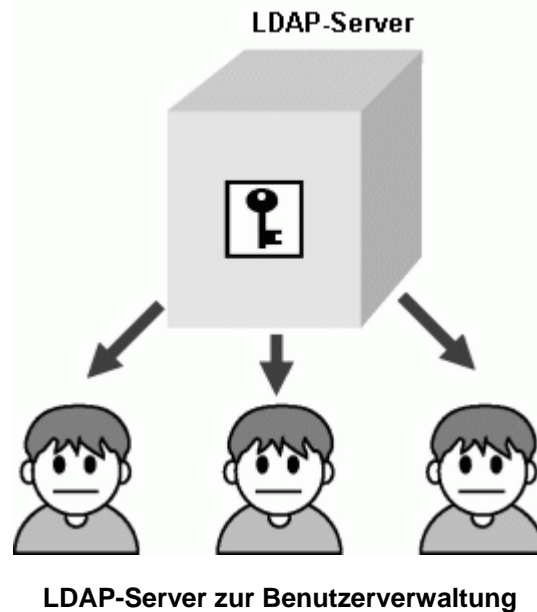
Dies hat dazu geführt, dass es in den meisten Netzen verschiedene Verzeichnisse mit teilweise redundanten Informationen gibt. Diese können oftmals nur schwer zusammen genutzt werden. Noch schwieriger bzw. fast unmöglich ist die Wartung dieser Daten. Ändert man etwas in dem einen Verzeichnis, so muss man es auch in allen Anderen ändern. So entstand der Wunsch nach zentraler Datenhaltung. Die Lösung heißt Verzeichnisdienst.

Verzeichnisse sind Bestandteil unseres täglichen Lebens. Jeder kennt das Konzept eines Verzeichnisses, wie z.B. das Telefonbuch. Im IT-Bereich wird aber statt dem Begriff Verzeichnis der Begriff Datenbank verwendet. Alltägliche Verzeichnisdienste im IT-Bereich sind u. a. DNS, Whois, LDAP.

---

<sup>1</sup> Verzeichnisdienst (engl. Directory Service) → Im Netzwerk verteilte Datenbank, die auf dem Client-Server-Prinzip basiert

LDAP stellt einen Verzeichnisdienst zur Verfügung, der zur Speicherung und Wiederabruf von Informationen einer einzelnen Person einer Organisation genutzt werden kann. Auf diese Weise können die verschiedensten Daten (Name, Telefonnummer, Daten von Benutzer-Accounts, usw.) gespeichert werden. Aktuell wird der LDAP-Dienst häufiger zum zentralen User-Management verwendet, als für andere Dinge.



### 1.3 Wo ist der Unterschied zwischen einem Verzeichnisdienst und einer konventionellen Datenbank?

Der Verzeichnisdienst ähnelt in seiner Arbeitsweise sehr einer normalen Datenbank wie MySQL. Auf den ersten Blick ist der Unterschied auch gar nicht so groß, die Besonderheit beim Verzeichnisdienst ist jedoch, dass dieser dahingehend optimiert ist, Suchvorgänge extrem schnell abzuschließen und die angefragten Daten dem Client zu übermitteln. Neueinträge im Verzeichnisdienst erfolgen allerdings deutlich langsamer. Zum häufigen Ändern von Daten ist der Verzeichnisdienst nicht geeignet. Der Schreibzugriff ist oftmals beschränkt auf die Administratoren.

Suchen funktioniert dafür umso besser und ist eine der Hauptoperationen eines Verzeichnisdienstes. Daher werden von dem Verzeichnisdienst ausführliche Suchfeatures zur Verfügung gestellt.

## 1.4 Wie kann LDAP die Arbeit in einem Netzwerk erleichtern?

Macht man sich einmal Gedanken darüber, wie viele Verzeichnisse in einem Netzwerk existieren, wird ziemlich schnell klar, dass die gespeicherten Daten redundant vorliegen (müssen). Bei Änderung der Daten, müssen diese an verschiedenen Stellen angepasst werden. Dies ist nicht sehr wartungsfreundlich und kann schnell im Chaos enden.

Auch sollte man bedenken, dass diese Verzeichnisse unter Umständen sensible Daten enthalten und somit an jeder Stelle die sichere Speicherung und Verarbeitung dieser gewährleistet sein muss (**z. B. TLS**).

Durch diese zentralisierte und normalisierte Speicherung der Daten und der besseren Konsistenz der Schnittstelle wird die Arbeit erheblich erleichtert.

## 1.5 Verwendete Software

In den folgenden Punkten wird der Zweck der eingesetzten Software kurz erläutert.

### 1.5.1 Berkeley DB

Die Berkeley Datenbank ist eine hochperformante Datenbank-Management-Software. Sie dient zur Ablage der Daten und stellt somit das Backend dar. Außerdem bietet sie eine Transaktionskontrolle, welche jedoch nicht unbedingt eingesetzt werden muss, da ein Verzeichnisdienst, wie es der LDAP darstellt, hauptsächlich auf Leseoperationen ausgelegt ist.

### 1.5.2 OpenLDAP

OpenLDAP stellt im Gegensatz zur Berkeley Datenbank das Frontend dar und ist für den Zugriff und die Organisation der Daten zuständig. OpenLDAP ist ein Verzeichnisdienst und auf das Lesen von Daten optimiert. Eine weiterführende Beschreibung ist unter den Punkten 1.1 bis 1.4 zu finden.

### 1.5.3 OpenSSL

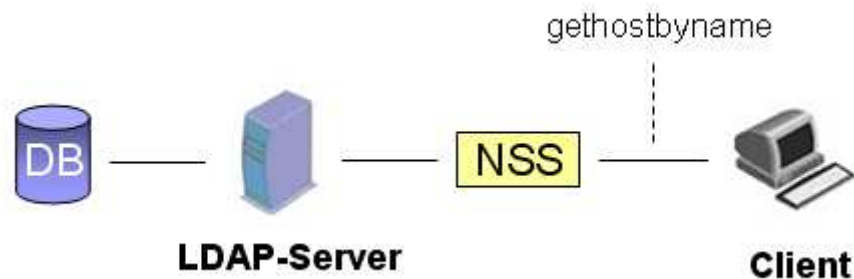
OpenSSL implementiert die Netzwerkprotokolle SSL (**S**ecure **S**ockets **L**ayer) und TLS (**T**ransport **L**ayer **S**ecurity). SSL ist ein Public-Key-Verschlüsselungsverfahren.

Bei der Verschlüsselung mit einem Public-Key-Verschlüsselungsverfahren werden zur Verschlüsselung und zur Entschlüsselung unterschiedliche Schlüssel verwendet. Diese Schlüsselpaare bestehen aus einem öffentlichen und einem privaten Schlüssel und werden von dem Empfänger der vertraulichen Nachrichten erzeugt. Der öffentliche Schlüssel wird vom Empfänger auf einem öffentlich zugänglichen Server hinterlegt. Wenn ihm jemand eine vertrauliche Nachricht zukommen lassen will, so nimmt er den öffentlichen Schlüssel aus dem Schlüsselspeicher und verschlüsselt damit die vertrauliche Nachricht. Diese Nachricht kann nun von niemandem, bis auf Ausnahme des Besitzers des privaten Schlüssels, entschlüsselt werden. Somit wird auch klar, warum der private Schlüssel gut verwahrt werden muss und keinem weitergegeben werden darf. Allerdings kann auch der Besitzer des privaten Schlüssels mit seinem privaten Schlüssel Nachrichten verschlüsseln. Diese Nachricht wäre dann von allen lesbar, die den öffentlichen Schlüssel besitzen. Somit wird sichergestellt, dass die Nachricht nur von einem best. Absender stammen kann, da sonst keiner den geheimen Schlüssel kennt.

Ein weiterer Anwendungsfall des Public-Key-Verfahrens ist die Authentifizierung von Usern. Kommuniziert man beispielsweise mit Partnern und möchte man deren Identität überprüfen, kann man dies über deren, von einer CA zertifizierten Zertifikate, tun.

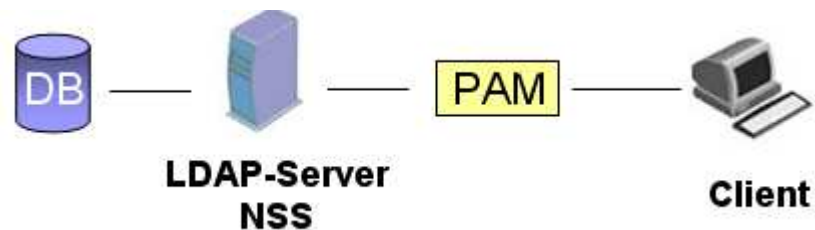
### 1.5.4 NSS\_LDAP

**NSS\_LDAP** ist eine Zwischenschicht zum LDAP-Server und stellt eine Schnittstelle zum Standard Name Service Switch dar. **NSS** stellt einem System sein Wissen bereit, welche unterschiedlichen Arten zur Namensauflösung zur Verfügung stehen und ermöglicht das Umschalten zwischen diesen. Ein Beispiel für eine Namensauflösung unter Linux ist die Datei `/etc/hosts`. In ihr werden Rechnernamen auf IP-Adressen abgebildet. Mit dem LDAP-Server sollen u.a. die Benutzer des Fachbereichs verwaltet werden.



### 1.5.5 PAM\_LDAP

**PAM** steht für **PluggableAuthenticationModules**. PAM übernimmt die Client-Funktion gegenüber dem Verzeichnisdienst und wird von den einzelnen Anwendungen aufgerufen. Es bietet eine Infrastruktur, die es Programmen ermöglicht, Benutzer über konfigurierbare Module zu authentifizieren. Das `pam_ldap`-Modul wird von der PAM-Einrichtung verwendet, um LDAP als Quelle von Benutzer-Authentifizierungs- und anderen Benutzer-Account-Daten zu nutzen. Zur Einbindung in PAM existieren verschiedene Module. Unter anderem für MySQL, Samba und natürlich auch LDAP.



### 1.5.6 SAMBA

Microsoft-Betriebssysteme nutzen für Netzwerkfreigaben das SMB-Protokoll (SMB → Server Message Block). Das SMB-Protokoll ist ein Protokoll für Datei-, Druck- und andere Serverdienste im Netzwerk unter Microsoft Windows-Betriebssystemen. Da Microsoft bisher mit jeder Betriebssystemversion das Protokoll überarbeitet hat, gibt es inzwischen sehr viele Protokollversionen und SMB-Dialekte. **Samba implementiert das SMB-Protokoll auf Unix-Maschinen.**

Seit der Version 2.0 kann Samba auch noch die Funktion des Domänencontrollers übernehmen. Eine Domäne ist im Prinzip eine Arbeitsgruppe von SMB-Computern, welche zusätzlich einen Domänencontroller besitzt. Die Funktionsweise des Domänencontrollers entspricht ungefähr dem eines NIS-Servers (Network Information Service), er verwaltet eine Datenbank über alle Benutzer- und Gruppeninformationen und führt auch ähnliche Dienste aus. Das heißt, er ist für viele sicherheitsrelevante Dinge zuständig. Er authentifiziert die Benutzer und überprüft ihre Berechtigung, auf Ressourcen zuzugreifen.

### 1.5.7 SMLDAP-Tools

Die SMLDAP-Tools sind ein Paket, bestehend aus verschiedenen Skripten, die das Verwalten von Benutzern und Benutzergruppen in LDAP über Samba und UNIX erleichtern.

## 2 Installation der benötigten Software

### 2.1 Berkeley Datenbank

Quelle: <http://www.sleepycat.com>

Version: 4.2.52, with encryption

Tar-Archiv entpacken und mit den folgenden Parametern kompilieren:

```
tar -xvzf db-4.2.52.tar.gz
cd db-4.2.52
cd build_unix
../dist/configure
make
make install
```

Alle Befehle, außer **make install** (muss als SuperUser ausgeführt werden), können als normaler User ausgeführt werden. **make** erzeugt lediglich die Binär-Files. **make install** kopiert zusätzlich noch die Dateien in die Systemverzeichnisse.

```
mv          /usr/include/db.h          /usr/include/db.bak
ln -s      /usr/local/BerkeleyDB.4.2/include/db.h /usr/include/
```

Mit der ersten Zeile wird die alte **db.h** gesichert. Sollten irgenwelche Probleme auftreten, kann man die alte Version wiederherstellen. **ln -s** fügt einen symbolischen Link zu der Datei **/usr/local/BerkeleyDB.4.2/include/db.h** in das Verzeichnis **/usr/include/** hinzu. Ein symbolischer Link verweist nicht direkt auf einen inode sondern auf einen Pfad zu einer Datei.

```
ldconfig -v
```

Liste der installierten Bibliotheken aktualisieren (**-v** steht für **verbose**).

## 2.2 OpenSSL

Quelle: <http://www.openssl.org>  
Version: 0.9.7d

```
tar -xvzf openssl-0.9.7d.tar.gz
cd openssl-0.9.7d
./config
make
make install
```

Auch hier muss **make install** wieder als SuperUser ausgeführt werden!

```
ldconfig -v
```

Liste der installierten Bibliotheken aktualisieren.

## 2.3 OpenLDAP

Quelle: <http://www.openldap.org>  
Version: 2.1.30

```
tar -xvzf openldap-stable-20040421.tgz
cd openldap-2.1.30
env LDFLAGS="-L/usr/local/BerkeleyDB.4.2/lib -L/usr/local/ssl/lib -L/usr/local/lib
-L/usr/lib"
CPPFLAGS="-I/usr/local/BerkeleyDB.4.2/include -I/usr/local/ssl/include"
./configure --sysconfdir=/etc
--enable-crypt
--without-cyrus-sasl
--enable-ipv6=no
--enable-bdb
--enable-ldap
--enable-slapd
--enable-syslog
--with-tls
--enable-wrappers
--without-kerberos

make depend
make
make test
make install
```

Mit **env** kann man Umgebungsvariablen für eine Programmausführung setzen. **LDFLAGS** (linker flags) spezifizieren den Pfad zu den library-Verzeichnissen, **CPPFLAGS** (C Preprocessor flags) zu den include-Verzeichnissen. **make test** führt lediglich eine Test-Kompilierung durch, so dass die richtige Installation nicht unvollständig abbricht und somit irgendwo unnütze Dateien im System abgelegt werden.

Leider ist bei unserer Installation das Problem aufgetreten, dass trotz der mit Hilfe der LDFLAGS eingebunden Libraries, die Installation an einer fehlenden Library

(**/usr/local/BerkeleyDB.4.2/lib**) gescheitert ist. Umgehen konnte man dieses Problem, indem man in die Datei `/etc/ld.so.conf` die Zeile mit dem Pfad zur Library einbindet, also **/usr/local/BerkeleyDB.4.2/lib** hinzufügt.

### Erläuterung der Kompile-Parameter

<code>--sysconfdir=/etc</code>	Standard-Pfad zu Konfigurationsdateien ändern, diese stehen dann in <code>/etc/openldap</code>
<code>--enable-crypt</code>	Unterstützung für verschlüsselte Passwörter
<code>--without-cyrus-sasl</code>	Cyrus SASL Unterstützung deaktivieren
<code>--enable-ipv6=no</code>	IPv6-Unterstützung ausschalten
<code>--enable-bdb</code>	Berkeley DB mit Transaktionskontrolle
<code>--enable-ldap</code>	LDAP-Backend aktivieren
<code>--enable-slapd</code>	slapd erstellen
<code>--enable-syslog</code>	Logdatei aktivieren, um evtl. auftretende Fehler identifizieren zu können
<code>--with-tls</code>	TLS/SSL Support
<code>--enable-wrappers</code>	Zum definieren von Zugriffsbeschränkungen und Filtern
<code>--without-kerberos</code>	Kerberos Unterstützung deaktivieren

## 2.4 NSS\_LDAP

Quelle: <http://www.padl.com>  
Version: 2.20

```
tar -xvzf nss_ldap.tgz.tar
cd nss_ldap-220
./configure
make
make install
```

Wenn Probleme auftreten, müssen evtl. noch mittels **export LDFALGS** bzw. **export CPPFLAGS** die richtigen Flags gesetzt werden.

## 2.5 PAM

Quelle: <http://www.pam.com>  
Version: 1.69

```
tar -xvzf pam_ldap.tgz.tar
cd pam_ldap-169
./configure --with-ssl
make
make install
```

### Erläuterung der Kompile-Parameter

--with-ssl	Unterstützung für SSL wird eingebunden
------------	--

## 2.6 Samba

Quelle: <http://www.samba.org>  
Version: 3.0.4

```
tar -xvzf samba-3.0.4.tar.gz
cd samba-3.0.4/source
./configure --sysconfdir=/etc/samba
             --localstatedir=/var/samba
             --with-smbwrapper
             --with-ldapsam
             --with-automount
             --with-quotas
             --with-acl-support
             --with-ssl
             --with-ssl-lib=/usr/local/ssl/lib
             --with-ssl-inc=/usr/local/ssl/include

make
make install
```

### Erläuterung der Kompile-Parameter

--sysconfdir=/etc/samba	Samba Konfigurationsdateien werden in diesem Verzeichnis abgelegt
--localstatedir=/var/samba	Verzeichnis, in dem die Protokolldatei abgelegt werden
--with-smbwrapper	ermöglicht Zugriff auf SMB-Freigaben
--with-ldap	Unterstützung für LDAP
--with-automount	NFS-Freigaben beim ersten Versuch der Zugriffs automatisch mounten
--with-quotas	Fügt Unterstützung für Quotas hinzu (experimentell)
--with-acl-support	Unterstützung für Windows NT/2000/XP-ACLs hinzufügen
--with-ssl	SSL-Unterstützung
--with-sslinc=/usr/local/ssl/lib	Pfad zu SSL-Libraries
--with-sslinclude=/usr/local/ssl/include	SSL-Include-Pfad

## 2.7 Installation erforderlicher Perlmodule

Für die Nutzung der SMLDAP Tools von *Idealx.org* müssen auf den meisten Systemen noch folgende Perlmodule installiert werden.

Quelle: [www.cpan.org](http://www.cpan.org)

### perl-ldap-0.31.tar.gz

```
tar -xvzf perl-ldap-0.31.tar.gz
cd perl-ldap-0.31
perl Makefile.PL
make
make install
```

### Authen-SASL-2.08.tar.gz

```
tar -xvzf Authen-SASL-2.08.tar.gz
cd Authen-SASL
perl Makefile.PL
make
make install
```

### Convert-ASN1-0.18.tar.gz

```
tar -xvzf Convert-ASN1-0.18.tar.gz
cd Convert-ASN1-0.0.18
perl Makefile.PL
make
make install
```

### **Crypt-SSLeay-0.51.tar.gz**

```
tar -xvzf Crypt-SSLeay-0.51.tar.gz
cd Crypt-SSLeay-0.51.tar.gz
perl Makefile.PL
make
make install
```

### **IO-Socket-SSL-0.96.tar.gz**

```
tar -xvzf IO-Socket-SSL-0.96.tar.gz
cd IO-Socket-SSL-0.96.tar.gz
perl Makefile.PL
make
make install
```

### **Net\_SSLeay.pm-1.25.tar.gz**

```
tar -xvzf Net_SSLeay.pm-1.25.tar.gz
cd Net_SSLeay.pm-1.25.tar.gz
perl Makefile.PL
make
make install
```

### **XML-Namespacesupport-1.08**

```
tar -xvzf XML-Namespacesupport-1.08
cd XML-Namespacesupport-1.08
perl Makefile.PL
make
make install
```

### **XML-SAX-0.12**

```
tar -xvzf XML-SAX-0.12
cd XML-SAX-0.12
perl Makefile.PL
make
make install
```

## 2.8 SambaLDAP-Tools

Quelle: <http://samba.idealx.org/index.en.html>  
Version: 0.8.5

```
tar -xvzf smbldap-tools-0.8.3.tar.gz
cd smbldap-tools

mkdir /etc/smbldap-tools
cp *.conf /etc/smbldap-tools
cp smbldap-* /usr/local/sbin/

chmod 644 /etc/smbldap-tools/smbldap.conf
chmod 600 /etc/smbldap-tools/smbldap-bind.conf

tar -xvzf mkntpwd.tar.gz
cd mkntpwd
make
make install
cp mkntpwd /usr/local/sbin/mkntpwd
```

## 2.9 JDK

Java wird zwar nicht direkt für das LDAP-Projekt benötigt, jedoch gibt es ein komfortables Tool, den LDAP-Browser, der in Java programmiert wurde.

Quelle: <http://www.java.sun.com>  
Version: j2sdk-1\_4\_2\_05-linux-i586.bin

Hat man die aktuelle Version heruntergeladen, muss man diese noch entpacken und in ein Verzeichnis verschieben, in welches man das Paket dauerhaft ablegen möchte.

```
sh j2sdk-1_4_2_05-linux-i586.bin
mv j2re1.4.2_05 /usr/local/java
```

Um nicht jedesmal, wenn Java benötigt wird, die Pfade zu Java neu setzen zu müssen, kann man dies automatisch vom System erledigen lassen. Dazu fügt man in die Datei /etc/profile folgende Zeilen ein:

```
JAVA_BINDER=/usr/local/java/bin
JAVA_HOME=/usr/local/java
JDK_HOME=/usr/local/java
JRE_HOME=/usr/local/java

PATH="$PATH:/usr/local/java/:/usr/local/java/bin/"

export JAVA_BINDER
export JAVA_HOME
export JDK_HOME
export JRE_HOME
```

```
export PATH
```

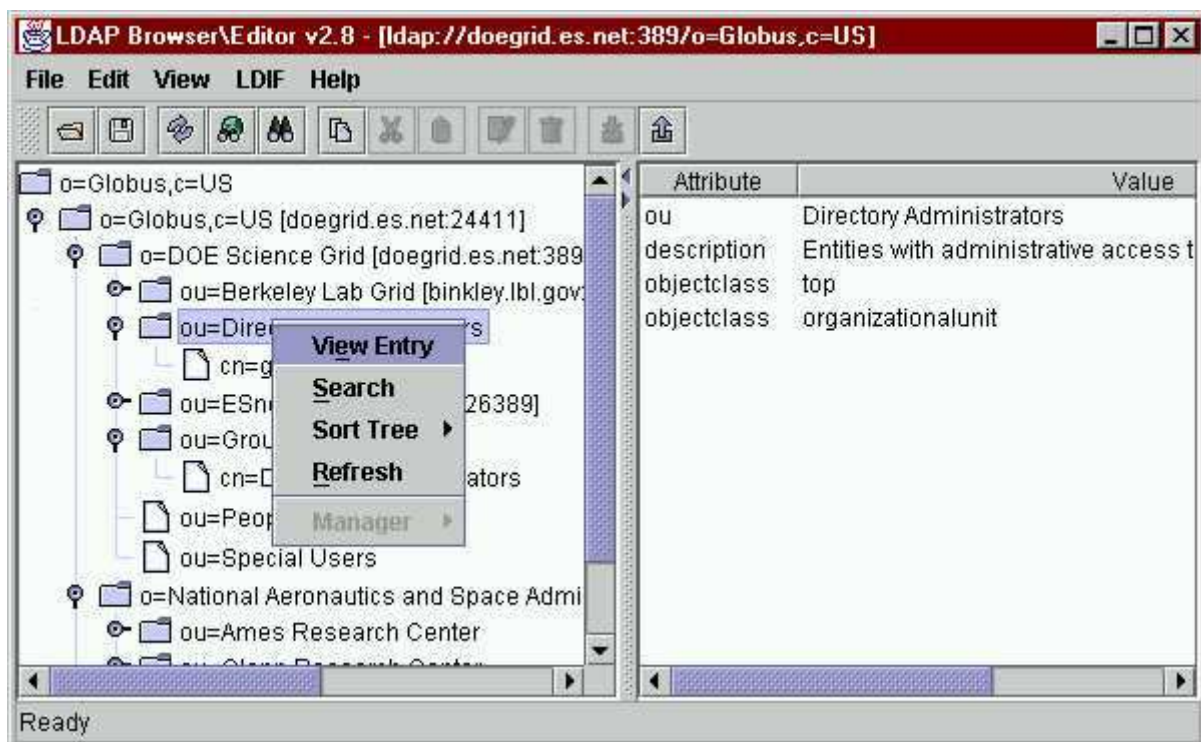
## 2.10 LDAP-Browser

Der LDAP-Browser ist ein komfortables Tool um ganz einfach, mittels GUI, Veränderungen an der Datenstruktur vorzunehmen.

Quelle: <http://www.iit.edu/~gawojar/ldap/>  
Version: 2.8.2 beta II

```
tar -xvzf Browser282b2.tar.gz  
cd ..  
cd ldapbrowser  
./lbe.sh
```

**lbe.sh** ist ein mitgeliefertes Startskript für den LDAP-Browser, der alle benötigten JAVA-Flags setzt und somit einen unkomplizierten Start ermöglicht.



## 3 Konfiguration der Komponenten

### 3.1 Berkeley DB

Keine weiterführende Konfiguration erforderlich.

### 3.2 OpenSSL

Keine weiterführende Konfiguration erforderlich.

### 3.3 OpenLDAP

Zunächst wird im Verzeichnis **/etc/init.d** ein Init-Skript für den LDAP-Server erstellt. (Quelle s. Appendix, /etc/init.d/openldap).

Damit der LDAP-Server bei Hoch- bzw. Herunterfahren des Systems automatisch gestartet / gestoppt wird, werden im Verzeichnis /etc/rc5 (RunLevel 5, Netzwerk) ein Startscript (S49openldap) und ein Stoppscript (K49openldap) erstellt.

**Anpassen** der Konfigurationsdatei des LDAP-Servers **slapd.conf**:

```
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/nis.schema
include      /etc/openldap/schema/inetorgperson.schema

schemacheck on

pidfile      /var/openldap/slapd.pid
argsfile     /var/openldap/slapd.args

database     bdb
suffix       "o=simpsons,c=de"
rootdn       "cn=Manager, o=simpsons,c=de "
rootpw       {crypt}s4L9sOIJo4kBM

directory   /usr/local/var/openldap-data
index        objectClass,uid      eq
```

Die ersten vier Zeilen sind zum Einbinden verschiedener **Schema-Dateien**. In diesen Schema-Dateien sind standardisierte Objekte definiert, um die Zusammenarbeit mit anderen Verzeichnisdienst-Servern zu gewährleisten. Ein LDAP-Schema definiert die Liste möglicher Typen von Einträgen (**Objektklassen**) zusammen mit den mit ihnen verknüpften **Attributen**. Diese Schema-Definitionen werden in Dateien abgespeichert. Objektklassen sind hierarchisch angeordnet, die Objektklasse **top** bildet die Wurzel der Baumstruktur.

Mittels **schemacheck on** wird überprüft, ob gültige Schema-Dateien vorliegen. Mögliche Fehler sind unter anderem, dass eine Vererbungshierarchie nicht eingehalten wurde (Endlosschleife).

Die beiden Einträge **pidfile** und **argsfile** sind für den laufenden Betrieb nötig (pid = Process id, args = Argumente). Für den LDAP-Server gibt es eine Prozesserkennung (PID), die in der angegebenen Datei protokolliert wird. Neben der PID wird zusätzlich eine Argument-Datei angelegt. In ihr sind Informationen zu den Startparametern des Servers zu finden.

Mit dem Eintrag **database** wird das zu verwendende Datenbankformat spezifiziert. Die **suffix**-Zeile gibt die Domäne an, für die der LDAP-Server Informationen bereitstellt. Gleichzeitig ist dieser Eintrag die Wurzel des LDAP-Verzeichnisses. **rootdn** gibt den selbstgewählten Namen des LDAP-Administrators an. **rootpw** ist das Passwort des LDAP-Administrators. Um es nicht im Klartext in die Datei slapd.conf schreiben zu müssen, kann man mit dem folgenden Befehl ein verschlüsseltes Passwort anlegen:

```
slappasswd -h {crypt}
```

Mit dem Parameter **directory** wird der Pfad zur Datenbank angegeben. Die Art der Indizierung in der Datenbank wird mit **index...** festgelegt (*index {<attrlist>|default} [pres,eq,approx,sub,<special>}*).

Standardmäßig erhält jeder Benutzer Lesezugriff auf die Datenbank.

### Anpassen der Client-Konfigurationsdatei **ldap.conf**:

```
### connection settings #####
host ldapsrv          # CN in Certificate
port 389
URI= ldap://ldapsrv:389/
#####

### pam settings #####
pam_filter objectclass=posixAccount
pam_login_attribute uid
#####

### nss settings #####
nss_base_passwd      o=auth_user,o=simpsons,c=de?sub
nss_base_shadow      o=auth_user,o=simpsons,c=de?sub
nss_base_group       o=auth_user,o=simpsons,c=de?sub
#####
```

**host** gibt die Adresse des LDAP-Servers an. Anstelle der IP-Adresse kann man auch den Namen des LDAP-Servers angeben, muss dann allerdings dafür sorgen, dass der Name aufgelöst wird. Mit **port** wird der Port angegeben, auf dem der LDAP-Server „lauscht“.

**base** spezifiziert die Wurzel des LDAP-Verzeichnisses.

Mit den zwei Zeilen zu den Einstellungen von **PAM**, wird PAM mitgeteilt, nach welchem Attribut(en) es Ausschau halten soll, um Benutzer zu authentifizieren.

Zur Konfiguration von NSS wird diesem noch mitgeteilt, unter welchem Zweig des „Datenbaumes“ die zu authentifizierenden Benutzer zu finden sind.

Zum Schluss werden noch evtl. doppelt vorhandene **ldap.conf**-Dateien überschrieben und ein Link auf die neu erstellte Datei gesetzt.

```
In -sf /etc/ldap.conf /etc/openldap/ldap.conf
```

### 3.4 NSS

**Anpassen** der Datei **/etc/nsswitch.conf**:

```
passwd: files, ldap  
shadow: files, ldap  
group: files, ldap  
hosts: files dns
```

### 3.5 Populieren der Datenbank

Mit folgendem Befehl wird die LDIF Datei in die Datenbank eingefügt:

```
slapadd -l struktur.ldif
```

Das Aufsplitten in mehrere ldif-Files ist hierbei sinnvoll, da man so Fehler besser lokalisieren kann.

Durchsucht werden kann die Datenbank mit

```
ldapsearch -x -D „dn=Manager,o=simpsons,c=de“ -W
```

### 3.6 Testen der bisherigen Konfiguration

Nachdem die Datenbank nun einige Einträge enthält, wird das ordnungsgemäße Zusammenwirken von nss, pam und LDAP getestet.

Um zu überprüfen, ob das System Benutzer aus unserer Datenbank erkennt, wird in der Shell das Kommando „**getent passwd**“ abgesetzt. Dabei sollten zusätzlich zu denen in der **/etc/passwd** eingetragenen Benutzer, die Benutzer aus dem LDAP-Verzeichnis angezeigt werden.

Werden unsere LDAP-User mit „getent passwd“ angezeigt, überprüfen wir die Funktionalität von PAM. Wir versuchen, uns auf unserem Server mittels „ssh“ mit einem LDAP-User einzuloggen:

```
ssh localhost -l ldapuser
```

...

### 3.7 NFS-Server konfigurieren

Um den Benutzern nach erfolgtem Login ihre Home-Verzeichnisse zur Verfügung zu stellen, muss ein **NetworkFileSystem-Server** (NFS) aufgesetzt werden, der die zentral gespeicherten Home-Verzeichnisse exportiert.

Da der NFS-Server beim Systemstart mitgestartet werden soll, wird ein Initskript in **/etc/init.d** und anschließend jeweils ein Start- und ein Stopskript in **/etc/rc5.d** angelegt.

Damit der Server das Home-Verzeichnis exportiert, wird die Datei **/etc/exports** folgendermaßen angepasst:

```
/home 192.168.0.0/255.255.255.0(rw)
```

In der Datei **/etc/exports** werden Verzeichnisse angegeben, die der NFS-Server exportieren soll. Zusätzlich wird angegeben, wer die Freigabe mounten darf und welche Rechte der Freigabe zugewiesen sollen (read-only oder read-write Rechte).

In unserem Beispiel wird das Verzeichnis **/home** des Servers für alle IP-Adressen im entsprechenden Subnet freigegeben. Die Clients erhalten sowohl Lese- als auch Schreibrechte.

### 3.8 NFS-Clients konfigurieren

Damit die vom NFS-Server freigegeben Verzeichnisse auf den Unix-Maschinen beim Systemstart gemountet werden, muss folgender Eintrag in der **/etc/fstab** getätigt werden:

```
192.168.0.200:/home /home nfs nodev,nolock 0 0
```

### 3.9 Samba konfigurieren

#### **/etc/samba/smb.conf**

```
[global]
netbios name = LDAP_PDC
workgroup = FBI
os level = 65
domain master = yes
local master = yes
preferred master = yes
security = user
domain logons = yes
log level = 1
log file = /var/samba/%m.log
encrypt passwords = yes
```

**netbios name** - mit netbios name wird der Name des Smbaservers innerhalb des Suchdienstes bestimmt. Dieser Eintrag ist nicht zwingend nötig, da ansonsten der Hostname des Systems benutzt wird. Allerdings kann innerhalb der Konfigurationsdatei der netbios name für das Abarbeiten von Skripts verwendet werden und sollte deshalb eingetragen werden.

**workgroup** - beschreibt den Namen, den die Windows Domäne bekommen soll. Der Eintrag heißt workgroup, da er im Normalfall den Arbeitsgruppennamen im Suchdienst beschreibt.

**os level** - beschreibt den OS Wert den Samba bei der Wahl zum Hauptsuchdienst angibt. Mit 65 sticht Samba damit auf jeden Fall alle aktuellen Betriebssysteme aus. Windows 2003 wird wahrscheinlich einen Wert von 64 haben.

**domain master** - macht Samba zum Domänen-Hauptsuchdienst.

**local master** - erlaubt es Samba an der Wahl des Domänen Hauptsuchdienstes teilzunehmen.

**preferred master** - erzwingt eine Hauptsuchdienstwahl beim Start des Samba Dienstes.

**security** - security legt die Authentifizierungsmethode des Clients fest. Der Eintrag kann vier verschiedene Werte haben: **share**, **user**, **server**, **domain**. Bei **share** werden wie bei Windows 9x Kennwörter auf einzelne Freigaben gesetzt. Bei **user** haben die einzelnen Benutzer einen Benutzernamen und ein Kennwort mit dem sie sich am Server authentifizieren. **Domain** steht für eine Windows Domäne, an der sich der User authentifiziert. Dabei ist nicht der aktuelle Samba als PDC beauftragt, sondern ein anderer Rechner innerhalb des Netzes. Bei **server** hat ein anderer Server die Kennwörter, dieser Server könnte mit dem password server Eintrag spezifiziert werden.

**domain logons** - domain logons veranlasst Samba Domänen-Anmeldungen auszuführen. Der Eintrag ist zur Ausführung eines PDC zwingend erforderlich.

**log level** - Synonym für debuglevel,

**log file** - gibt Pfad und Datei an, in der logging-Informationen gespeichert werden

**encrypt passwords** - sorgt für verschlüsselte Kennwörter. Dies ist zwingend erforderlich, wenn Samba als Domänencontroller eingesetzt werden soll.

```
passdb backend = ldapsam:ldap://ldapsrv
ldap admin dn = uid=Sambaroot,o=auth_user,o=simpsons,c=de
ldap suffix = o=simpsons,c=de
ldap machine suffix = o=auth_user
ldap user suffix = o=auth_user
ldap group suffix = ou=Groups
```

**passdb backend** - passdb backend legt fest, wie die Benutzernamen und Kennwörter gespeichert werden. Wird die Option nicht angegeben wird per Default **tdbsam** verwendet. In unserem Fall wird der LDAP-Server als Backend verwendet. Zu beachten ist, dass der Hostname (hier: ldapsrv) durch einen DNS oder die lokale /etc/hosts aufgelöst werden muss.

**ldap admin dn** - gibt den **DistinguishedName** an, mit dem Samba den LDAP-Server kontaktiert

**ldap suffix** - gibt die Basis der benötigten Daten in der Datenhierarchie an

**ldap machine suffix** - gibt den Ort an, an dem Informationen zu Computerkonten gespeichert werden. In Kombination Samba - LDAP (aktuelle Versionen) scheint hier ein Bug vorzuliegen. Beim Erstellen neuer Maschinenkonten bzw. beim Überprüfen bestehender Konten sucht Samba nicht an der angegebenen Stelle. Aus diesem Grund werden vorübergehend die Maschinenkonten ebenfalls in der Gruppe „People“ abgespeichert.

**ldap user suffix** - gibt den Ort an, an dem die Benutzerinformationen gespeichert sind.

**ldap group suffix** - gibt den Ort an, an dem Gruppen und Gruppeninformationen gespeichert sind.

```
logon drive = Z:  
logon path = \\%L\profiles\%U  
logon home = \\LDAP_PDC\%U  
  
logon script = logon.bat  
  
passwd program = /usr/local/sbin/smbldap-passwd %u  
add machine script = /usr/local/sbin/smbldap-useradd -w "%u"  
add user script = /usr/local/sbin/smbldap-useradd -m "%u"
```

**logon drive** - Home-Laufwerke unter Windows bedeuten das Selbe wie unter Unix. Es sind Verzeichnisse, die nur dem Benutzer zugänglich sind und welche zentral auf einem Server gespeichert werden. Somit ist es leicht möglich, Datensicherungen durchzuführen. Wir möchten nun unter Windows automatisch unser Home-Laufwerk erhalten, dazu müssen zwei Einträge in der **smb.conf** getätigt werden. In der **[global]** Sektion wird ein **logon drive** gesetzt. Anschließend wird in der **[homes]** Sektion definiert, welche Eigenschaften der Freigabe zugeordnet werden sollen.

**logon path** - Um Samba für Roaming Profiles zu konfigurieren, müssen wir Samba mitteilen, wo die Profile abgelegt werden sollen. Logon Path definiert den Ablageort für die Profile, dabei steht %L für den NetBIOS-Namen des aktuellen Servers und %u für den Usernamen. Der Pfad steht in Abhängigkeit der Freigabe **profiles**, die in der Sektion **[profiles]** definiert wird.

**logon home** - gibt die Lokation der Home-Verzeichnisse der Benutzer an

**logon script** - Mit Hilfe der Logon-Scripts werden, in Abhängigkeit des Users, unterschiedliche lokale Einstellungen getätigt oder Programme gestartet. Oftmals werden Laufwerksmapping durchgeführt oder die lokale Zeit des Rechners synchronisiert.

**passwd program** - Hier wird das Programm angegeben, dass für Passwortänderungen verwendet werden soll. Damit sowohl Windows-Passwörter als auch Unix-Passwörter zugleich geändert werden, kommen hier die SMBLDAP-Tools zum Einsatz.

**add machine script** - Sorgt für das Einfügen von Computern in die Domäne des Samba Servers. Dies ist nötig, da das Domänen Konzept von Microsoft richtigerweise Computer und Benutzer kennt und diese getrennt voneinander in der SAM bzw. im LDAP-Server hält. Um die Computer nun bei Bedarf, ohne manuelles Anlegen, in der Domäne zu registrieren, wird ein Unix Account ohne Login-Shell und ohne Home-Verzeichnis erzeugt. Damit das Hinzufügen eines Computeraccounts bzw. Benutzeraccounts reibungslos funktioniert, wird hier das Programm „**smbldap-useradd**“ verwendet.

**add user script** - siehe add machine script, sorgt für das Hinzufügen neuer Benutzer in die Domäne.

```
[netlogon]
comment = Samba Network Logon Service
path = /etc/samba/netlogon
public = no
read only = yes
writable = no
browseable = no
```

Damit Samba tatsächlich als PDC agieren kann, verlangen die Windows Clients eine Freigabe namens „**netlogon**“ auf dem Domänen Controller, mit der sie sich bei ihrem Domainlogon verbinden können.

**[netlogon]** - wie man sieht, wird der Freigabename immer als Sektionsüberschrift vorangestellt.

**path** - beschreibt den absoluten Pfad der Freigabe.

**writable** - besagt, ob auf der Freigabe geschrieben werden kann oder nicht.

**browsable** - erlaubt das Anbieten einer Freigabe in Suchlisten. Unter Windows wird dies mit einem abschließenden Dollarzeichen gemacht.

Die Freigabe Netlogon existiert also, ist schreibgeschützt und wird nicht in der Suchliste beim Durchforsten des Netzes angegeben. Somit ist die Grundkonfiguration unseres Samba-Servers abgeschlossen. Im Prinzip könnten wir ihn jetzt als PDC benutzen.

```
[homes]
comment = U% home directory
browseable = no
guest ok = no
writable = yes
```

Man sieht, dass der Pfad der Freigabe nicht gesetzt ist. Samba sucht den Pfad des Home-Laufwerkes aus der **/etc/passwd** und ordnet es dem Benutzer zu.

**guest ok** - sagt aus, ob es sich um eine public Freigabe handelt oder nicht. Mit **no** kann nur der betreffende User darauf zugreifen.

```
[profiles]
path=/home/samba/profiles
browseable=no
writeable = yes
create mask = 600
directory mask = 0700
read only = no
guest ok = yes
```

### 3.10 SambaLDAP-Tools konfigurieren

Die SMBLDAP-Tools greifen auf zwei Konfigurationsdateien zurück:

1. **/etc/smbldaptools/smbldap\_bind.conf**
2. **/etc/smbldaptools/smbldap.conf**

In der Datei **/etc/smbldap\_bind.conf** werden Angaben gemacht, unter welchem Namen und mit welchem Passwort Änderungen in der Datenbank getätigt werden sollen. (rootdn in slapd.conf).

```
slaveDN="cn=Manager,o=simpsons, c=de"
slavePw="ldapuser"

masterDN="cn=Manager,o=simpsons, c=de"
masterPw="ldapuser"
```

Da in unserem Projekt nur ein Server zum Einsatz kommt, sind die Einträge für Master- und SlaveDN identisch. Es müssen beide Einträge getätigt werden.

**masterDN / slaveDN** - entspricht dem Manager, der in der slapd.conf angegeben wurde.

**masterPw / slavePw** - entspricht dem Passwort, dass in slapd.conf als rootpw angegeben wurde. **Achtung:** Das Passwort sollte nicht im Klartext gespeichert werden!

#### **/etc/smbldaptools/smbldap.conf**

```
# General Settings

UID_START="1000"
GID_START="1000"

SID="S-1-5-21-1885275206-936062365-1706894471"
```

**UID\_START** - wird ein Benutzer mit „**smbldap-useradd**“ erstellt, erhält der erste Benutzer die UserId 1000, der Zweite die 1001, usw.

**GID\_START** - wird eine Gruppe mit „**smbldap-groupadd**“ erstellt, so erhält die erste Gruppe die GruppenId 1000, die Zweite die 1001, usw.

**SID** - domain secure ID, wird mit „**net getlocalsid**“ ermittelt.

```
# LDAP Configuration
slaveLDAP="192.168.0.200"
slavePort="389"
masterLDAP="192.168.0.200"
masterPort="389"
ldapTLS="0"
suffix="o=simpsons,c=de"
usersdn="o=auth_user,o=simpsons,c=de"
computersdn="o=auth_user,o=simpsons,c=de"
groupsdn="o=auth_group,o=simpsons,c=de"
```

**slaveLDAP** - IP-Adresse / Hostname des LDAPSlave-Servers, wird kein Slave-Server eingesetzt, wird hier der gleiche Eintrag getätigt, der unter masterLDAP getätigt wurde. Soll später die Kommunikation über SSL/TLS stattfinden, muss hier unbedingt der CommonName (CN) eingetragen werden, der beim Erstellen des Serverzertifikates angegeben wurde.

**slavePort** - Port, auf dem der LDAP-SLAVE-Server läuft.

**masterLDAP** - IP-Adresse / Hostname des LDAP-MASTER-Servers. Soll später die Kommunikation über SSL/TLS stattfinden, muss hier unbedingt der CommonName (CN) eingetragen werden, der beim Erstellen des Serverzertifikates angegeben wurde.

**masterPort** - Port, auf dem der LDAP-MASTER-Server läuft.

**ldapTLS** - soll die Kommunikation zwischen den SMLDAP-Tools und LDAP-Server nicht über TLS stattfinden, wird der Wert auf „0“ gesetzt, ansonsten auf „1“.

**suffix** - gibt die Basis der benötigten Daten in der Datenhierarchie an.

**usersdn** - gibt an, wo Benutzerinformationen gespeichert werden sollen.

**computersdn** - gibt an, wo Maschinenkonten gespeichert werden sollen.

**groupsdn** - gibt an, wo Gruppeninformationen gespeichert werden sollen.

```
# Unix Accounts Configuration
userLoginShell="/bin/bash"
userHomePrefix="/home/"
defaultUserGid="513"
defaultComputerGid="513"
```

**userLoginShell** - Voreinstellung für neue Benutzer, als Shell wird die „**bash**“ verwendet

**userHomePrefix** - Verzeichnis, in dem die Home-Verzeichnisse der Benutzer gespeichert werden

**defaultUserGid / defaultComputerGid** - Standard-GruppenID für neue Benutzer / Computer

```
# SAMBA Configuration
userSmbHome=""
userProfile=""
userHomeDrive=""
userScript=""
```

In diesem Abschnitt werden allen Parameter ein NULL-String zugewiesen. Dadurch werden die Parameter aus der Konfigurationsdatei von SAMBA (**smb.conf**) verwendet.

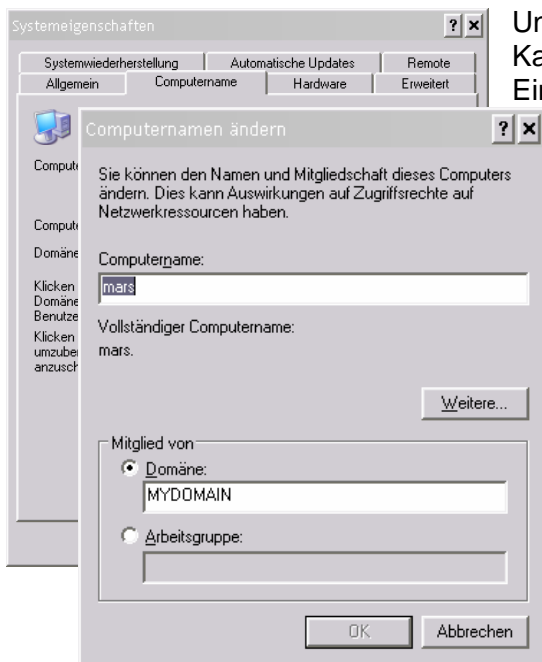
```
# SMLDAP-TOOLS Configuration
with_smbpasswd="0"
smbpasswd="/usr/local/samba/bin/smbpasswd"
mk_ntpasswd="/usr/local/sbin/mkntpwd"
```

**with\_smbpasswd** - es wird empfohlen, **with\_smbpasswd** auf "0" zu setzen, damit nicht das Passwort-Programm von Samba verwendet wird, sondern ein angepasstes Programm „mk\_ntpasswd“.

**mk\_ntpasswd** - Pfadangabe zum Programm.

### 3.11 Testen der Konfiguration, Hinzufügen Windows-Rechners in die Domäne

Nachdem unser Domänen Controller jetzt Einsatzbereit ist, geht es um die Anbindung von Clients an diese Domäne. In unserem Testszenario waren es zwei Windows Rechner, ein Windows 2000 und ein Windows XP. Die Anbindung an eine bestehende Windows Domäne ist sehr einfach. Als erstes muss der Rechner in die Domäne eingebracht werden.



Unter Start → Systemsteuerung → System auf der Karteikarte Computernamen, sehen wir die aktuellen Einstellungen des Computers. Mit dem Knopf „Ändern“ können wir diese Einstellungen ändern. Bei „Mitglied von“ wird im Feld Domäne einfach der Name der Domäne, den wir in der smb.conf unter dem workgroup Eintrag abgelegt haben, eingetragen.

Mit einem Klick auf OK werden wir nach einem administrativen Zugang gefragt und zur Eingabe eines Benutzernamens und Kennworts eines Administrators aufgefordert. Hier können alle Benutzer, die unter admin users in der smb.conf stehen, sowie alle Mitglieder der Domain Admin Gruppe, angegeben werden. Bei erfolgreichem Abschluss befindet sich der Rechner nun in der Domäne und jedes Mitglied der Domäne kann sich nach einem Neustart an der Domäne anmelden. Dabei muss noch darauf geachtet werden, dass beim Anmelden, im Feld

"Anmelden an:", der Domänen Name angegeben ist. Eine lokale Anmeldung am System ist immer noch möglich. Falls WINS verwendet werden soll, so muss dieser in den Netzwerkeinstellungen der Clients ebenfalls eingetragen werden.

## 4 Das System mit SSL absichern

Um eine gesicherte Kommunikation zwischen dem LDAP-Server und dem Samba-Server sowie zwischen LDAP-Server und Client-Programmen zu realisieren, haben wir uns dazu entschieden, TLS über den Port 389 zu benutzen. Dazu müssen zunächst ein eigenes TrustCenter (CA), Server- und Client-Zertifikate erstellt werden.

### 4.1 CA (Certificate Authority)

Diese Instanz stellt das Bindeglied zwischen den Kommunikationspartnern dar. Beide Partner müssen praktisch dieser Instanz vertrauen und besitzen deren Zertifikat. Der private Schlüssel der CA muss besonders geschützt werden. Bevor eine CA ein Zertifikat zertifiziert und somit als vertrauenswürdig einstuft, werden das Zertifikat und die eingetragenen Daten sorgfältig geprüft. Das Zertifikat der CA wird an verschiedenen Stellen veröffentlicht, um seine Gültigkeit zu überprüfen. Findet ein Verbindungsaufbau statt, so verifizieren beide Seiten die übermittelten Zertifikate mithilfe des CA-Zertifikates. Verifizieren bedeutet hierbei, zu prüfen, ob die digitale Signatur der CA unter dem Zertifikat des Kommunikationspartners korrekt ist. Man kann Zertifikate also als elektronische Ausweise ansehen.

### 4.2 Erstellen des Root-Zertifikates (CA)

#### 4.2.1 Vorbereitungen für OPENSSL

```
mkdir certs csr datas keys private datas/ca.db.certs
touch private/ca.key datas/ca.db.serial
echo '01' > datas/ca.db.serial
cp /dev/null datas/ca.db.index
```

Pseudo-random bytes generieren

```
openssl rand 1024 > datas/random-bits
```

#### 4.2.2 Erstellen des Schlüssels der CA

Es wird nach einem Passwort (pass phrase) gefragt. Dieses Passwort muss jedes Mal eingegeben werden, wenn ein Zertifikat signiert werden soll.

```
openssl genrsa -des3 -out private/ca.key 1024 -rand datas/random-bits
chmod 600 private/ca.key
```

#### 4.2.3 Signieren der CA

```
openssl req -new -x509 -days 3650 -key private/ca.key -out certs/ca.pem
```

#### 4.2.4 Erstellen des LDAP-Server-Schlüssels

```
openssl genrsa -out keys/ldapsrv.key 1024
```

#### 4.2.5 Erstellen des LDAP-Server-Zertifikates

**Achtung:** Wenn nach dem CommonName (CN) gefragt wird, sollte der FullQualifiedDomainName (FQDN) angegeben werden. Unter diesem Namen muss der LDAP-Server erreichbar sein.

```
openssl req -new -key keys/ldapsrv.key -out csr/ldapsrv.csr
```

#### 4.2.6 Signieren des LDAP-Server-Zertifikates mit der CA

```
openssl ca -config ca.conf -out certs/ldapsrv.txt -infiles csr/ldapsrv.csr
```

```
perl -n -e 'm/BEGIN CERTIFICATE/ && do {$$seen=1}; $$seen && print;' < certs/ldapsrv.txt  
> certs/ldapsrv.pem
```

Das Zertifikat kann nun verifiziert werden:

```
openssl verify -CAfile certs/ca.pem certs/ldapsrv.pem
```

#### 4.2.7 Erstellen des Schlüssels für SMBLDAP-Tools

```
openssl genrsa -out keys/smbldap-tools.key 1024
```

#### 4.2.8 Erstellen des Zertifikates für SMBLDAP-Tools

**Achtung:** Wenn nach dem CommonName (CN) gefragt wird, sollte „smbldap-tools“ angegeben werden.

```
openssl req -new -key keys/smbldap-tools.key -out csr/smbldap-tools.csr
```

#### 4.2.9 Signieren des SMBLDAP-Tools-Server-Zertifikates mit der CA

```
openssl ca -config ca.conf -out certs/smbldap-tools.txt -infiles csr/smbldap-tools.csr
```

```
perl -n -e 'm/BEGIN CERTIFICATE/ && do {$$seen=1}; $$seen && print;' < certs/smbldap-  
tools.txt > certs/smbldap-tools.pem
```

Das Zertifikat kann nun verifiziert werden:

```
openssl verify -CAfile certs/ca.pem certs/smbldap-tools.pem
```

#### 4.3 Erstellen der Zertifikate für LDAP-Clients

Siehe „Erstellen des Schlüssels für SMBLDAP-Tools“. Als CN muss wiederum der Hostname (FQDN) angegeben werden.

#### 4.4 Anpassen der Konfigurationsdatei "slapd.conf"

```
### SSL Konfiguration #####  
TLSCertificateFile      /etc/openldap/ldapsrv.pem  
TLSCertificateKeyFile   /etc/openldap/ldapsrv.key  
TLSCACertificateFile    /etc/openldap/ca.pem  
TLSCipherSuite :SSLv3  
TLSVerifyClient demand  
  
# verbietet nicht-verschlüsselte Verbindungen zum LDAP-Server auf port 389  
# wenn security tls=1  
security tls=1  
#####
```

#### 4.5 Anpassen der Konfigurationsdatei „smbldap-tools.conf“

```
# Use TLS for LDAP  
# If set to 1, this option will use start_tls for connection  
# (you should also used the port 389)  
ldapTLS="1"  
  
# How to verify the server's certificate (none, optional or require)  
# see "man Net::LDAP" in start_tls section for more details  
verify="none"  
  
# CA certificate  
# see "man Net::LDAP" in start_tls section for more details  
cafile="/etc/openldap/ca.pem"  
  
# certificate to use to connect to the ldap server  
# see "man Net::LDAP" in start_tls section for more details  
clientcert="/etc/smbldap-tools/smbldap-tools.pem"  
  
# key certificate to use to connect to the ldap server  
# see "man Net::LDAP" in start_tls section for more details  
clientkey="/etc/smbldap-tools/smbldap-tools.key"
```

#### 4.6 Anpassen der Konfigurationsdatei "ldap.conf", (Client)

```
###  tls settings  #####  
TLS_CHECKPEER yes  
TLS_CACERT      /etc/certs/ca.pem  
TLS_CERT        /etc/certs/client1.pem  
TLS_KEY          /etc/certs/client1.key  
ssl start_tls  
#####
```

#### 4.7 Anpassen der Konfigurationsdatei "smb.conf"

```
ldap ssl = start_tls
```

## 5 Appendix

### 5.1 OpenLDAP als Adressbuch für Outlook bzw. andere eMail-Clients

Die folgende Tabelle zeigt einige, von verschiedenen Adress-Verwaltungsprogrammen, genutzte Parameter. Leider können bei der Nutzung einige Probleme auftreten. Outlook 2003 beispielsweise nutzt initials, um die Initialen einer Person darzustellen. In dem Standard Windows Adressbuch, welches auch von Outlook Express genutzt wird, wird initials für den zweiten Vornamen verwendet.

Des Weiteren fragen die verschiedenen Adress-Programme unterschiedliche Parameter ab und verwenden diese auch auf unterschiedliche Art. Hier besteht momentan leider noch keine einheitliche Schnittstelle zum LDAP-Server. Die Parameter die vom jeweiligen Programm benutzt, bzw. abgefragt werden, können bei der Debug-Ausgabe des LDAP-Servers abgelesen werden. Danach hilft nur noch "googeln" oder ausprobieren weiter, um herauszufinden, welcher Parameter was bedeutet.

Werden Einträge beim Hinzufügen in das LDAP-Verzeichnis beanstandet, müssen evtl. die schema-Dateien um die betreffenden Attribute erweitert werden, bzw. komplett neue schema-Dateien erstellt werden.

c, co, countryName	Land (dienstlich)	core.schema
department, ou, organizationalUnitName	Abteilung	core.schema
display-name, cn, commonName	Name, der angezeigt wird	core.schema
facsimileTelephoneNumber	Faxnummer (dienstlich)	core.schema
givenName	Vorname	core.schema
initials	2. Vorname / Initialen	core.schema
l	Stadt (dienstlich)	core.schema
mail	eMail-Adresse	core.schema
organizationName, o	Firmenname	core.schema
physicalDeliveryOfficeName	Büro (dienstlich)	core.schema
postalAddress, streetAddress, street	Straße (dienstlich)	core.schema
postalCode	PLZ (dienstlich)	core.schema
sn	Nachname	core.schema
surname	Alternativer Nachname <sup>1</sup>	core.schema
telephoneNumber	Telefonnummer (dienstlich)	core.schema
title	Position	core.schema
homePhone	Telefonnummer (privat)	cosine.schema
homePostalAddress	Straße (privat)	cosine.schema
mobile	Handynummer (privat)	cosine.schema
pager	Pager	cosine.schema
comment	Anmerkung	officePerson.schema
conferenceInformation	Server für Netmeeting	officePerson.schema
IPPhone	IP-Telefon	officePerson.schema
otherFacsimileTelephoneNumber	Faxnummer (privat)	officePerson.schema
otherPager	Alternativer Pager	officePerson.schema

#### Von Adressbüchern genutzte Attribute

Um die oben aufgelisteten Parameter nutzen zu können, müssen die entsprechenden schema-Dateien in der **slapd.conf** Datei eingebunden werden.

<sup>1</sup> Der zweite Nachname wird weder in diversen getesteten Windows-Adressbüchern noch in diversen getesteten Linux-Adressbüchern angezeigt. Da die Suchparameter (Filter) der Adressbuch-Programme nicht auswählbar sind, ist es schwer, nach alternativen Parametern zu suchen. In diesem Fall könnte man in diesem alternativen Nachnamen den Status eines Mitglieds der FH speichern (Student, Professor, ...). Somit wäre es relativ leicht möglich nach best. Gruppen zu suchen.

Außerdem müssen in die Einträge, die im LDAP-Server gespeichert werden sollen, die entsprechenden objectClass-Einträge hinzugefügt werden (inetOrgPerson, officePerson). Eine Suchabfrage des Standard-Adressbuches von Windows sieht z. B. folgendermaßen aus:

```
filter: ((mail=max*)((cn= max *)((sn= max *)(givenName= max *))))
```

Dieser logische Ausdruck gibt an, dass alle eMail-Adressen, alle Anzeigenamen, alle Vornamen und alle Nachnamen die mit Max anfangen angezeigt werden sollen.

Die oben gezeigte Tabelle zeigt eine Auswahl von Parametern, die von Outlook genutzt werden. Da andere Programme andere Parameter verwenden, ist es nicht möglich, Listen für alle möglichen Programme auszustellen. Im Internet sind allerdings gute Parameterlisten zu finden.

Momentan scheint die Verwendung von OpenLDAP als Adressbuch nur bedingt geeignet, da keine einheitliche Schnittstelle existiert. Außerdem erhält man nicht wirklich ein Adressbuch, sondern man kann nur nach best. Attributen suchen und bekommt dann eine Liste der gefundenen Einträge als Ergebnis.

## 5.2 OpenLDAP

### 5.2.1 /etc/rc5.d/S49openldap

```
In -s /etc/init.d/openldap /etc/rc5.d/S49openldap
```

### 5.2.2 /etc/rc5.d/K49openldap

```
In -s /etc/init.d/openldap /etc/rc5.d/K49openldap
```

### 5.2.3 /etc/openldap/slapd.conf

```
### Benötigte Schemata einbinden #####
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/samba.schema
include /etc/openldap/schema/openldap.schema
include /etc/openldap/schema/officeperson.schema
#####

### SSL Konfiguration #####
TLSCertificateFile /etc/openldap/ldap.pem
TLSCertificateKeyFile /etc/openldap/ldap.key
TLSCACertificateFile /etc/openldap/ca.pem
TLSCipherSuite :SSLv3
TLSVerifyClient demand
# verbietet nicht-verschlüsselte Verbindungen zum LDAP-Server auf port 389
security tls=1
#####
```

```
#####
schemacheck on
pidfile /var/openldap/slapd.pid
argsfile /var/openldap/slapd.args
#####

###   ldbm database definitions   #####
database bdb
suffix "o=simpsons,c=de"
rootdn "cn=Manager,o=simpsons,c=de"
# To generate a hash of your password so that it is not in plain text
# in this document use the command:
# slappasswd -s myldappassword
rootpw ldapuser
#####

# Store the database in the directory /usr/local/var/openldap-data
directory /usr/local/var/openldap-data
#####

###   Indices to maintain   #####
index objectClass,uid,uidNumber,gidnumber,ou,memberUid eq
index cn,mail,surname,givenname eq,subinitial
index sambaSID,sambaPrimaryGroupSID,sambaDomainName eq
#####

###   ACL definieren   #####
access to attrs=userPassword,sambaLMPassword,sambaNTPassword
    by dn="uid=Sambaroot,o=auth_user,o=simpsons,c=de" write
    by * auth

access to attrs=loginShell
    by * read
    by self write

access to dn.children="o=auth_user,o=simpsons,c=de"
    attrs=objectClass,sambaSamAccount
    by dn="uid=Sambaroot,o=auth_user,o=simpsons,c=de" write
    by * read

access to dn.children="ou=Hosts,o=simpsons,c=de"
    attrs=objectClass,sambaSamAccount
    by dn="uid=Sambaroot,o=auth_user,o=simpsons,c=de" write
    by * read

access to dn.children="ou=ldmap,o=simpsons,c=de"
    by dn="uid=Sambaroot,o=auth_user,o=simpsons,c=de" write
    by * read

access to dn.subtree="o=simpsons,c=de"
    by dn="uid=Sambaroot,o=auth_user,o=simpsons,c=de" write
    by * read
#####
```

### 5.2.4 /etc/ldap.conf (Client)

```
### Connection Settings #####
host ldapsrv
port 389
URI= ldap://ldapsrv:389/
#####

### Pam Settings #####
pam_filter objectclass=posixAccount
pam_login_attribute uid
#####

### NSS Settings #####
nss_base_passwd o=auth_user,o=simpsons,c=de?sub
nss_base_shadow o=auth_user,o=simpsons,c=de?sub
nss_base_group o=auth_user,o=simpsons,c=de?sub
#####

### TLS Settings #####
TLS_CHECKPEER yes
TLS_CACERT /secure/certs/ca.pem
ssl path /secure/certs/
TLS_CERT /etc/certs/client1.pem
TLS_KEY /etc/certs/client1.key
ssl start_tls
#####
```

### 5.3 /etc/nsswitch

```
passwd: files, ldap
shadow: files, ldap
group: files, ldap
hosts: files dns
```

## 5.4 /etc/pam.d/login

```
#PAM
auth required pam_nologin.so
auth sufficient pam_ldap.so
auth sufficient pam_unix.so shadow use_first_pass
auth required pam_deny.so
auth requisite pam_unix2.so nullok          #set_secrcp
auth required pam_securetty.so
auth required pam_nologin.so
auth required pam_env.so
auth required pam_mail.so
account required pam_unix2.so
password required pam_pwcheck.so nullok
password required pam_unix2.so nullok use_first_pass use_authtok
session required pam_unix2.so none # debug or trace
session required pam_limits.so
```

## 5.5 SAMBA

### 5.5.1 /etc/samba/smb.conf

```
[global]
netbios name = LDAP_PDC
workgroup = FBI

# Instruct Samba to store the passwords in your LDAP-Server
# ldapsrc entspricht dabei exakt dem Namen, der bei der Erstellung
# des Serverzertifikates als Common Name angegeben wurde
passdb backend = ldapsam:ldap://ldapsrv

# This defines which backend to use for group id mappings
idmap backend = ldap:ldap://ldapsrv

# LDAP Configuration
ldap admin dn = uid=Sambaroot,o=auth_user,o=simpsons,c=de
ldap delete dn = no
ldap ssl = start tls
# ldap ssl = ssl

# Define where are the machine accounts stored in your LDAP-Server
ldap suffix = o=simpsons,c=de
ldap machine suffix = o=auth_user
ldap user suffix = o=auth_user
ldap group suffix = ou=Groups
ldap idmap suffix = ou=ldmap
os level = 65
domain master = yes
local master = yes
preferred master = yes
security = user
domain logons = yes
```

```
# Mount user's home directories in drive Z:
logon drive = Z:

# Define the location of each user's profile
logon path = \\LDAP_PDC\profiles\%U

# Define each users home dir
logon home = \\LDAP_PDC\%U

# Use the logon script named MachineArchitecture.bat
logon script = logon.bat
log level = 1
log file = /var/samba/%m.log
password level = 4
encrypt passwords = yes
passwd chat = *New*password* %n\n *Retype*new*password* %n\n
              *all*authentication*tokens*updated*
passwd program = /usr/local/sbin/smbldap-passwd %u
ldap passwd sync = Yes
add user script = /usr/local/sbin/smbldap-useradd -m "%u"
add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
add group script = /usr/local/sbin/smbldap-groupadd -p "%g"
add user to group script = /usr/local/sbin/smbldap-groupmod -m "%u" "%g"

[netlogon]
comment = Samba Network Logon Service
path = /etc/samba/netlogon
public = no
read only = yes
writable = no
write list = support
browseable = no

[homes]
comment = U% home directory
browseable = no
guest ok = no
writable = yes

[profiles]
path=/home/samba/profiles
browseable=no
writeable = yes
create mask = 600
directory mask = 0700
read only = no
guest ok = yes
```

## 5.6 SSL

### 5.6.1 /secure/ca.conf

```
default_ca = default_CA

[default_CA]
dir = . # Where everything is kept
certs = ./certs # Where the issued certs are kept
new_certs_dir = ./datas/ca.db.certs # Where the issued crl are kept
database = ./datas/ca.db.index # database index file
serial = ./datas/ca.db.serial # The current serial number
RANDFILE = ./datas/random-bits # private random number file
certificate = ./certs/ca.pem # The CA certificate
private_key = ./private/ca.key # The private key
default_days = 730
default_crl_days = 30
default_md = md5
preserve = no
x509_extensions = server_cert
policy = policy_anything

[policy_anything]
countryName = optional
stateOrProvinceName = optional
localityName = optional
organizationName = optional
organizationalUnitName = optional
commonName = supplied
emailAddress = optional

[server_cert]
#subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
extendedKeyUsage = serverAuth,clientAuth,msSGC,nsSGC
basicConstraints = critical,CA:false
```

## 5.7 SambaLdapTools

### 5.7.1 /etc/smbldap-tools/smbldap.conf

```
#####  
# General Configuration  
#####  
  
# UID and GID starting at...  
UID_START="1000"  
GID_START="1000"  
# Put your own SID  
# to obtain this number do: net getlocalsid  
SID="S-1-5-21-1885275206-936062365-1706894471"  
  
#####  
# LDAP Configuration  
#####  
  
# Ex: slaveLDAP=127.0.0.1  
slaveLDAP="ldapsrv"  
slavePort="389"  
  
# Master LDAP : needed for write operations  
# Ex: masterLDAP=127.0.0.1  
masterLDAP="ldapsrv"  
masterPort="389"  
  
# Use TLS for LDAP  
# If set to 1, this option will use start_tls for connection  
# (you should also used the port 389)  
ldapTLS="1"  
  
# How to verify the server's certificate (none, optional or require)  
# see "man Net::LDAP" in start_tls section for more details  
verify="none"  
  
# CA certificate  
# see "man Net::LDAP" in start_tls section for more details  
cafile="/etc/smbldap-tools/ca.pem"  
  
# certificate to use to connect to the ldap server  
# see "man Net::LDAP" in start_tls section for more details  
clientcert="/etc/smbldap-tools/smbldap-tools.pem"  
  
# key certificate to use to connect to the ldap server  
# see "man Net::LDAP" in start_tls section for more details  
clientkey="/etc/smbldap-tools/smbldap-tools.key"  
  
# LDAP Suffix  
# Ex: suffix=dc=IDEALX,dc=ORG  
suffix="o=simpsons,c=de"  
  
# Where are stored Users  
# Ex: usersdn="ou=Users,dc=IDEALX,dc=ORG"
```

```
usersdn="o=auth_user,o=simpsons,c=de"

# Where are stored Computers
# Ex: computersdn="ou=Computers,dc=IDEALX,dc=ORG"
computersdn="o=auth_user,o=simpsons,c=de"

# Where are stored Groups
# Ex groupsdn="ou=Groups,dc=IDEALX,dc=ORG"
groupsdn="o=auth_group,o=simpsons,c=de"

# Default scope Used
scope="sub"

# Unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA)
hash_encrypt="SSHA"

#####
# Unix Accounts Configuration
#####

# Login defs
# Default Login Shell
# Ex: userLoginShell="/bin/bash"
userLoginShell="/bin/bash"

# Home directory prefix (without username)
# Ex: userHomePrefix="/home/"
userHomePrefix="/home/"

# Gecos
userGecos="System User"

# Default User (POSIX and Samba) GID
defaultUserGid="513"

# Default Computer (Samba) GID
defaultComputerGid="513"

# Skel dir
skeletonDir="/etc/skel"

#####
# SAMBA Configuration
#####

# The UNC path to home drives location without the username last extension
# (will be dynamically prepended)
# Ex: \\My-PDC-netbios-name\homes
# Just set it to a null string if you want to use the smb.conf 'logon home'
# directive and/or disabling roaming profiles
userSmbHome=""

# The UNC path to profiles locations without the username last extension
```

```
# (will be dynamically prepended)
# Ex: \\My-PDC-netbios-name\profiles\
# Just set it to a null string if you want to use the smb.conf 'logon path'
# directive and/or disabling roaming profiles
userProfile=""

# The default Home Drive Letter mapping
# (will be automatically mapped at logon time if home directory exist)
# Ex: q(U:) for U:
userHomeDrive=""

# The default user netlogon script name
# if not used, will be automatically username.cmd
# make sure script file is edited under dos
userScript=""

#####
# SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
#####

# Allows not to use smbpasswd (if with_smbpasswd == 0 in smbldap_conf.pm) but
# prefer mkntpwd... most of the time, it's a wise choice :- )
with_smbpasswd="0"
smbpasswd="/usr/local/samba/bin/smbpasswd"
mk_ntpasswd="/usr/local/sbin/mkntpwd"
```

### 5.7.2 /etc/smbldap-tools/smbldap\_bind.conf

```
# secure, link: /etc/smbldap-tools/smbldap_bind.conf --> /src/config/smbldap_bind.conf.sec

#####
# Credential Configuration
#####

# Notes: You can specify two differents configuration if you use a
# master ldap for writing access and a slave ldap server for reading access
# By default, we will use the same DN (so it will work for standard Samba release)

slaveDN="cn=Manager,o=simpsons,c=de"
slavePw="ldapuser"

masterDN="cn=Manager,o=simpsons,c=de"
masterPw="ldapuser"
```

## 6 Fazit

Wenn man erst einmal einigermaßen durch LDAP und seine benötigten Komponenten durchblickt, ist es tatsächlich möglich, mit OpenLDAP und Komponenten eine Single-Sign-On-Lösung relativ einfach zu realisieren :-). Der Weg dorthin ist jedoch ziemlich steinig und kann einige Nerven kosten.

Im Internet sind kaum brauchbare Anleitungen zu diesem Thema zu finden. Zwar gibt es viele Anleitungen, diese sind jedoch recht oberflächlich. Versucht man nach diesen Anleitungen ein System zu installieren und zu konfigurieren, fällt schnell auf, dass die Verfasser der Anleitungen wahrscheinlich noch nie in ihrem Leben einen LDAP-Server aufgesetzt haben.

Die Dokumentationen der Hersteller zu den einzelnen verwendeten Komponenten hätte auch besser sein können. Leider scheint die Dokumentation von Freeware eine nicht so große Rolle zu spielen bzw. scheint die Aufgabe der Dokumentation unter den einzelnen Entwicklern teilweise sehr unbeliebt zu sein.

Letztendlich ist es uns in mühevoller Kleinarbeit doch noch gelungen, die benötigten Konfigurationen auszuarbeiten, so dass es möglich sein sollte, nach unserer Anleitung eine funktionsfähige Single-Sign-On-Lösung zu realisieren.

Es lässt sich aber nicht vermeiden, dass durch die Vielzahl der erhältlichen Distributionen immer noch bei einzelnen Anwendern bei der Installation und Konfiguration Fehler auftreten. Leider ist es nicht möglich alle diese Fehlerkonstellationen zu berücksichtigen. Teilweise liegt es an fehlenden Komponenten, die bei der Betriebssystem-Installation nicht mitinstalliert (make) werden, teilweise können auch fehlende Pfad-Angaben zum Fehler führen. Die Liste der möglichen Fehler lässt sich nahezu unendlich weiterführen.

Will man einen LDAP-Server und seine Komponenten auf einem PC einsetzen, mit dem aktiv Benutzer arbeiten, sollte man auf jeden Fall eine Image seiner Festplatte anlegen, so dass man im Fehlerfall seine alte Konfiguration wieder zur Verfügung hat.

In unserem konkreten Fall haben wir uns leider mehrfach bei der Installation der LDAP-Komponenten unser Betriebssystem „zerschossen“. In Folge dessen musste das System komplett neu aufgesetzt werden.

Das LDAP-Projekt war zwar ziemlich arbeitsintensiv und die Nächte waren dadurch sehr kurz, trotzdem hat es viel Spaß gemacht.

## 7 Quellen

<http://www.idealx.org>  
<http://www.mitlinux.de/ldap/>  
<http://linux.cudeso.be/linuxdoc/ldap.php>  
<http://www.linuxhaven.de/dlhp/HOWTO/DE-LDAP-HOWTO.html>  
<http://139.18.184.171/howtos/nss-pam-ldap/>  
<http://www.openldap.org/doc/admin22/>  
<http://www.mykleines.de>  
<http://www.linux-magazin.de/Artikel/ausgabe/1998/09/LDAP/ldap.html>  
<http://www.europe.redhat.com/documentation/rhl6.2/ref-guide-de/s1-ldap-redhattips.php3>  
<http://www.coding-board.de/board/forumdisplay.php?f=55>

<http://www.openldap.org/faq/data/cache/294.html>  
<http://www.yo-linux.com/TUTORIALS/LinuxTutorialLDAP-GILSchemaExtension.html>

<http://ivs.cs.uni-magdeburg.de/~dumke/ProSem/faust.html>  
<http://www.math.gatech.edu/~dijuremo/ldap/cacontent.html>