LDAP-Server Setup

für den Privatgebrauch

Aldo Brießmann

14.04.2021
(überarbeitet am 30.04.2021)

Übersicht

Disclaimer

- dieser Vortrag basiert auf Recherchen und Experimenten und wird professionellen Ansprüchen möglicherweise nicht genügen
- Hauptziel ist ein Setup für den Privatgebrauch
- hier wird nur OpenLDAP behandelt. Andere LDAP-Implementierungen unterscheiden sich stellenweise
- LDAP ermöglicht viel, ist aber nicht immer besonders einstiegsfreundlich

Wann ist LDAP sinnvoll?

Szenario	**LDAP (o.ä.) sinnvoll?**
einzelner Dienst	eher nein
mehrere Dienste (und/oder mehrere User)	je nach Zeit und Motivation
viele Dienste, viele User	wahrscheinlich ja

was ist LDAP?

LDAP: **Lightweight Directory Access Protocol**

Def. Directory Service/Verzeichnisdienst

Ein Dienst, der in einem Netzwerk Daten bestimmter Art zur Verfügung stellt.

- Daten können verglichen, gesucht, erstellt, modifiziert und gelöscht werden
- OpenLDAP ist angelehnt an x.500-Standard zum Aufbau eines Verzeichnisdienstes

mögliche Einsatzzwecke

- Authentifizierung und Bereitstellung von Gruppen für Dienste
  - v.a. Web-Dienste, aber auch z.B. Samba/smb
- Benutzerverwaltung für Linux-Systeme (insbesondere für mehrere Rechner gleichzeitig)
- *(Adressbuch für Thunderbird und Outlook: **nicht getestet**, erfordert zusätzliche Schemata. Für Firmen oder große gemeinsame Kontaktverzeichnisse interessant, für private Adressbücher gibt es einfachere Lösungen.)*

Vorteile

- nur ein Username+Passwort für mehrere Dienste
- neue Dienste können schnell für alle User verfügbar gemacht werden
- POSIX-Benutzerkonten-Features für User möglich (z.B. Passwort-Ablauf, Ort des Home-Verzeichnisses, ...) über PAM-Modul `pam_ldap`
- optionale fortgeschrittene Features: SSH-Key(s) für User, Mailadressen für Postfix

Wissenswertes

- LDAP definiert Inhalt und ermöglicht Zugriff auf Daten über Schemata
- viele Implementierungen: **OpenLDAP**, Microsoft Active Directory, ApacheDS, Redhat Directory Server, ...[1]
- Konfiguration und Definition heute standardmäßig in Datenbank selbst
  - früher in externer Config-Datei `/etc/ldap/slapd.conf`; ist deprecated, wird aber von den meisten Tutorials noch genutzt

[1] siehe `https://ldap.com/directory-servers/`

Geschichte von LDAP

- 1990: x.500-Standard veröffentlicht
- 1992: erste LDAP-Implementierung von der Universität Michigan (ab 1995 mit SLAPD/Standalone LDAP Daemon)
- 1995: LDAPv2 standardisiert, seit 2003 offiziell veraltet
- 1997: LDAPv3 standardisiert (bis heute in Gebrauch)
- 1998: OpenLDAP-Projekt basierend auf Michigan-LDAP-Implementierung begonnen
- heute: OpenLDAP 2.4.58 mit LDAPv3
- nahe Zukunft: OpenLDAP 2.5 (seit Oktober 2020 als Beta verfügbar)

Aufbau Teil I: Abkürzungen und Fachbegriffe

DIB (Directory Information Base)

Information eines LDAP-Servers; *Gruppen und User*

DIT (Directory Information Tree)

Definition der hierarchischen **Struktur** der Einträge in der DIB

DUA (Directory User Agent)

"Benutzer"/**Client**-System, das Anfragen stellen kann, z.B. Web-Dienst mit LDAP-Authentifizierung

DSA (Directory System Agent)

LDAP-**Server**-Dienst, empfängt und beantwortet Anfragen

Aufbau Teil II: Schema

- **Schema:** Set von Definitionen und Beschränkungen für die Struktur des DIT. *z.B. mögliche Namen, Informationen im Eintrag, Attribute, Zugriffsberechtigungen, ...*[2]

Formulierung in ABNF (Augmented Bacchus-Naur Form)

Schema-Definition PosixAccount [3]

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

[2] siehe RFC4512

[3] siehe https://oidref.com/1.3.6.1.1.1.2.0

Aufbau Teil II: Schema

ObjectClasses legen Eigenschaften eines LDAP-Objekts fest: Name, Beschreibung, Superklasse, Art, verpflichtende und optionale Attribute

- drei verschiedene Arten: Abstract, Structural, Auxiliary
- Structural können alleine genutzt werden, Auxiliary und Abstract benötigen andere ObjectClass(es)
- ein LDAP-Objekt hat eine oder mehrere ObjectClasses, deren Definitionen es dann erfüllen muss
- ObjectClasses können nachträglich hinzugefügt und entfernt werden, solange dabei weiterhin alle Bedingungen erfüllt sind

Beispiele: account, posixAccount, posixGroup, alias, ...

Aufbau Teil III: Datenbank(en)

allgemein: zwei interne Datenbanken (frontend "*{-1}*", config "*{0}*") und eine (oder mehrere) für eigentliche Nutzdaten "*{1}*", "*{2}*", ...
Die Konfigurationsdateien liegen unter `/etc/ldap/slapd.d/cn=config/`, die eigentliche Datenbank unter `/var/lib/ldap/`. NICHT die Dateien dort bearbeiten, immer LDAP-Tools nutzen!

Datenbank-Typen:

- BDB:[4] Oracle Berkley DB. Veraltet und ineffizient
- HDB:[4] Hierarchical DB. Variante von BDB, aber ein bisschen effizienter
- MDB: Memory-Mapped DB. Neues Speicherformat, weniger manuelle Einstellschrauben, aktuell empfohlen

[4] DEPRECATED, wird in OpenLDAP 2.5 entfernt

Aufbau Teil IV: Inhalte

Häufige Abkürzungen in LDAP-Einträgen:

- **dn:** distinguished name
- **dc:** domain component
- **cn:** common name
- **ou:** organizational unit
- **o:** organization

LDIF

LDAP Data Interchange Format, Dateiendung `.ldif`

LDIF Beispiel einzelner Eintrag

```
# tux, Users, mydomain.com
dn: uid=tux,ou=Users,dc=mydomain,dc=com
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
objectClass: simpleSecurityObject
gidNumber: 1001
uidNumber: 1001
homeDirectory: /home/tux
cn: tux
# Passwort-Hash generieren geht interaktiv mit dem
    Kommando `slappasswd`
userPassword:: {SSHA}OHxSnVV0K5JPRoKO5HxTHktTE49BPCNW
uid: tux
```

LDIF

mehrere Einträge in einer LDIF-Datei:

LDIF mehrere Einträge

```
1 dn: cn=tux,ou=Users,dc=mydomain,dc=com
2 objectClass: account
3 cn: tux
4 userPassword:: {SSHA}ZoDmQO5AV4esJUr/6KLE3G8I6htwr6po
5
6 dn: cn=wilber,ou=Users,dc=mydomain,dc=com
7 objectClass: account
8 cn: wilber
9 userPassword:: {SSHA}CQgKB9s0Qpj6CPdQKEoc3AqAkcj+r4p5
```

LDIF

LDIF für Modifikationen (insbesondere für `ldapmodify`):

LDIF mehrere Änderungen

```
1  dn: cn=tux,ou=Users,dc=mydomain,dc=com
2  changetype: modify
3  replace: userPassword
4  userPassword:: {SSHA}bhd4hgMk1lthrFxwAlPuO5oNhNoBGB47
5  -
6  replace: cn
7  cn: supertux
8  -
9  add: description
10 description:: Ein springender Pinguin
```

Setup Vorbereitung

hier für Debian; bei manchen anderen Distributionen ähnlich

- OpenLDAP-Dienst installieren:

```
$ sudo apt-get install slapd
```

- empfohlene Zusatzpakete: **ldap-utils** (für ldapadd, ldapmodify, ...), **ldapvi** (VIM-basierter DB-Editor)
- weitere interessante Pakete: **phpldapadmin** (Web-Interface), **Apache Directory Studio** (GUI, für Desktop-PC)

Setup über Autokonfiguration in Debian

```
$ sudo dpkg-reconfigure slapd
```

- omit server configuration: no
- DNS domain name: ‘mydomain.com‘ (was man will, idealerweise eigene Domain; wird dann zu dc=mydomain,dc=com zerlegt)
- Organization name: ‘myorganization‘
- administrator password: ‘********‘
  - etwas möglichst sicheres, da Vollzugriff möglich
  - bei global zugänglichem LDAP-Server genauer mit Absicherung beschäftigen!
  - Admin ist in Debian ‘cn=admin,dc=mydomain,dc=com‘
- Database Backend: ‘MDB‘
- remove DB when slapd is purged?: ‘no‘
- (falls LDAP-Installation schon mal lief:) Move old database?: ‘yes‘ (zum Wegsichern)

Initialen LDAP-Tree erstellen

Datei `anfangsbaum.ldif` mit folgendem Inhalt erstellen:

anfangsbaum.ldif

```
dn: ou=Users,dc=mydomain,dc=com
changetype: add
objectClass: top
objectClass: organizationalUnit
ou: Users

dn: ou=Groups,dc=mydomain,dc=com
changetype: add
objectClass: top
objectClass: organizationalUnit
ou: Groups
```

Organizational Units für Personen und Gruppen erstellen:

```
$ sudo ldapmodify -D cn=admin,dc=mydomain,dc=com -W
-f anfangsbaum.ldif
```

- `-D`: bind-dn (=Benutzername)
- `-W`: Passwort, interaktiv abgefragt
- `-f`: danach angegebene Datei nutzen
- optional: `-n`: "dry-run", zeigt nur an was getan werden würde

Danach z.B. mit `sudo slapcat` Datenbank-Inhalte ausgeben, um zu sehen, ob die Änderungen übernommen wurden.

Nutzer erstellen

allgemein:

- strukturelle ObjectClass `account`
- weitere sinnvolle ObjectClass: `posixAccount`
- normale User am besten im zuvor erstellten Users-Pfad
- per LDIF (nächste Folie) oder mit anderen Tools möglich
- Bearbeiten und Löschen von Nutzerkonten auf gleichem Weg

Nutzer per LDIF erstellen

firstuser.ldif

```
dn: uid=tux,ou=Users,dc=mydomain,dc=com
changetype: add
objectClass: account
objectClass: posixAccount
objectClass: top
cn: Tux
gidNumber: 1001
uidNumber: 1001
homeDirectory: /home/tux
uid: tux
# Hash für userPassword mit 'slappasswd' erstellt
userPassword:: {SSHA}6xtIwoXNYj3dDuj6GJbfPuxQD+Zs6Fe6
```

```
$ sudo ldapmodify -D cn=admin,dc=mydomain,dc=com -W
-f firstuser.ldif
```

Gruppen erstellen

- analog zu Benutzererstellung
- wichtig: es gibt verschiedene Arten von Gruppen (posixGroup, GroupOfNames, GroupOfUniqueNames), die von unterschiedlichen Programmen akzeptiert werden. Leider sind die Gruppen nicht miteinander kompatibel, weil sie unterschiedliche Attribute (memberUid, member, uniqueMember) nutzen

Gruppe per LDIF erstellen

`.ldif` mit folgendem Inhalt erstellen:

firstgroup.ldif

```
1       dn: uid=linuxgroup,ou=Groups,dc=mydomain,dc=com
2       changetype: add
3       objectClass: posixGroup
4       objectClass: top
5       cn: mylinuxgroup
6       gidNumber: 1001
7
```

Gruppe hinzufügen mit:

```
$ sudo ldapmodify -D cn=admin,dc=mydomain,dc=com -W
-f firstgroup.ldif
```

Nutzer zu Gruppen hinzufügen

- einfach gesagt: Gruppe ergänzen um Eintrag/Zeile mit entsprechendem Attribut, Wert ist der `dn` des Users, z.B. `uid=tux,ou=Users,dc=mydomain,dc=com`; Attribute sind in diesem Fall nicht einzigartig
- GroupOfNames und GroupOfUniqueNames brauchen zwingend zur Erstellung bereits mindestens einen User

Dienst-User erstellen

Datei `.ldif` mit folgendem Inhalt erstellen:

serviceuser.ldif

```
dn: cn=Search,dc=mydomain,dc=com
changetype: add
objectClass: top
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: Search
userPassword:: {SSHA}ZNtDQROQPuHkrrFAylDOrxelXJGkVDhn
```

User hinzufügen wieder mit:

```
$ sudo ldapmodify -D cn=admin,dc=mydomain,dc=com -W
-f serviceuser.ldif
```

Datenbankeinträge löschen

- über ldapdelete:

```
$ sudo ldapdelete 'uid=test123,dc=mydomain,dc=com'
```

  oder mit -f aus LDIF-Datei analog zu ldapmodify-Aufrufen
- über ldapmodify
- über andere Tools: üblicherweise einfach Eintrag löschen, ggf. auch noch aus Gruppen

Überblick

- Zugriff: per Bind, ggf. anonym; standardmäßig unverschlüsselt ohne SSL/TLS!
- binddn vs. simpleauth:
  - binddn benötigt ein Service-Nutzerkonto
  - simpleauth macht den Zugriff über das Userkonto
- viele Dienste synchronisieren einfach ihre interne Datenbank mit LDAP, Verzögerungen sind aber unüblich
- Gruppen können üblicherweise auch synchronisiert werden, werden aber unterschiedlich genutzt (GroupOfNames vs. GroupOfUniqueNames)

Parameter für Konfiguration bei einem Dienst

üblicherweise anzugeben:

- Host und Port des LDAP-Servers/-Dienstes
- bind dn (für Lesezugriff auf Nutzerdatenbank) und Passwort
- base dn: Pfad, unter dem alle Benutzerkonten liegen
- Benutzer- und/oder Admin-Filter: Anforderungen an Objekte, die als Benutzer gelten sollen, z.B. `(&(objectClass=posixAccount)(uid=%s))` (bei %s wird vom Programm der Anmelde-Name eingefügt)
- Attribute: Zuordnung von LDAP-Attributen zu Eigenschaften im Dienst, z.B. "uid soll als Anmeldename genutzt werden, mail als Mailadresse"

- LDAPS: verschlüsselter LDAP-Verkehr über Port 636, erfordert Zertifikat[5]
- Replication: für Ausfallsicherheit
- Overlays[6] können genutzt werden, um weitere Funktionen bereitzustellen, z.B. *ppolicy* Overlay für Password-Policies oder *memberof* Overlay für Reverse Group Membership (effizientere Gruppen-Abfrage für Benutzer)
- für größere Setups oder zum Basteln: mehrere Instanzen, die jeweils einen Teilbaum der gesamten LDAP-Datenbank haben

[5] siehe `https://www.openldap.org/doc/admin24/tls.html`
[6] siehe `https://openldap.org/doc/admin24/overlays.html`

RFCs:

- x.500: RFC1487
- LDAPv2: RFC1777
- LDAPv3: siehe RFC4510

Dokumentationen:

- https://www.openldap.org/doc/
- https://ldap.com/
  - LDAP-Fehlercodes: https://ldap.com/ldap-result-code-reference/
- https://ldapwiki.com/
- https://de.wikipedia.org/wiki/Verzeichnisdienst
- https://www.openldap.org/faq/data/cache/1.html
- Schemata: https://ldap.com/understanding-ldap-schema/
- https://www.digitalocean.com/community/tutorials/understanding-the-ldap-protocol-data-hierarchy-and-entry-components

Linux-Distributions-Wikis:

- https://wiki.debian.org/LDAP/
- https://wiki.archlinux.org/index.php/OpenLDAP
- https://old-en.opensuse.org/OpenLDAP/Basic_setup

verschiedene Howtos:

- OpenLDAP mit Thunderbird (von 2006, ungetestet!): https://www.pro-linux.de/artikel/2/106/openldap-adressbuch-fuer-thunderbird.html
- PAM: https://wiki.debian.org/LDAP/PAM
- Bind (DNS): https://www.flomain.de/2018/09/using-linux-with-openldap-for-user-dhcp-and-dns/
- Samba: https://wiki.samba.org/index.php/Samba_&_LDAP
- Howto LDIF: https://www.digitalocean.com/community/tutorials/how-to-use-ldif-files-to-make-changes-to-an-openldap-system