

17 Lösungsansätze mit Overlays

Im Abschnitt 11.1.2 auf Seite 87 wurden Overlays vorgestellt, deren Einsatz dann auch in unterschiedlichen Konfigurationsbeispielen vorgestellt wird. In diesem Kapitel beschreiben wir nun einige Aufgaben und deren Lösungsansätze, die mit einem Overlay realisiert werden können, aber bisher noch nicht beschrieben wurden.

17.1 Dynamische Einträge erstellen, darstellen und verwalten

Stellen wir uns folgendes Szenario vor: In einem Unternehmen werden die Netzwerkadressen durch einen DHCPD dynamisch vergeben, die Leases werden in einem Verzeichnisdienst als *dynamic Entries* hinterlegt. Als Attributtyp für die Internetadresse wird *ipHostNumber* verwendet. Die Aufgabe ist nun, festzustellen, welche Hosts zu einer gegebenen Zeit im Netz vorhanden sind.

Die Lösung kann mit den beiden Overlays *dds* und *dynlist* realisiert werden. Das Overlay *dds* wurde schon in Abschnitt 11.4.3 auf Seite 104 beschrieben. Zuerst soll einmal das Ergebnis präsentiert werden:

```
ldapsearch -LLL -Y digest-md5 -H ldap://localhost:9004 \
  -b cn=myDynamicHosts,o=avci,c=de -s base \
(objectclass=groupOfUris) iphostnumber
SASL/DIGEST-MD5 authentication started
Please enter your password:
SASL username: dieter
SASL SSF: 128
SASL data security layer installed.
dn: cn=dynamicGroup,o=avci,c=de

dn: cn=myDynamicHosts,o=avci,c=de
ipHostNumber: 192.168.100.31
ipHostNumber: 192.168.100.32
ipHostNumber: 192.168.100.33
ipHostNumber: 192.168.100.56
```

Das ist z.B. mit ISC-DHCP und der Erweiterung für LDAP-Support möglich, obwohl dann andere Attributtypen benutzt werden.

Listing 17.1
Suchergebnis

Und dies ist das Suchergebnis eines Host-Eintrags:

```
ldapsearch -LLL -Y digest-md5 -H ldap://localhost:9004 \
-b ou=myhosts,o=avci,c=de -s one cn=klm "*" +
SASL/DIGEST-MD5 authentication started
Please enter your password:
SASL username: dieter
SASL SSF: 128
SASL data security layer installed.
dn: cn=klm,ou=myHosts,o=avci,c=de
uid: klm
cn: klm
ipHostNumber: 192.168.100.56
objectClass: account
objectClass: ipHost
objectClass: dynamicObject
entryTtl: 7200
entryExpireTimestamp: 20070620193636Z
structuralObjectClass: account
entryUUID: 89b7e07c-b3a0-102b-9faa-77ec14ffb1df
creatorsName: cn=admin,o=avci,c=de
createTimestamp: 20070620173636Z
entryCSN: 20070620173636.022652Z#000000#000#000000
modifiersName: cn=admin,o=avci,c=de
modifyTimestamp: 20070620173636Z
entryDN: cn=klm,ou=myHosts,o=avci,c=de
subschemaSubentry: cn=Subschema
hasSubordinates: FALSE
```

Aus diesem Listing wird ersichtlich, dass vier Hosts zur Zeit online sind, Host `cn=klm` mit der Adresse `192.168.100.56` hat eine *Time To Live* von 7200 Sekunden und wird um 19:36 UTC wieder entfernt; dies sagen uns die operationalen Attribute *entryTtl* und *entryExpireTimestamp*.

Der Eintrag `ou=myHosts,o=avci,c=de` muss natürlich vorhanden sein, die einzelnen Host-Einträge werden zur Laufzeit dynamisch hinzugefügt, es dürfen also keine statischen Einträge sein.

Die `slapd.conf` enthält die im folgenden Listing gezeigten Parameter:

```
Listing 17.2 include /etc/openldap/schema/core.schema
dynamic Host include /etc/openldap/schema/cosine.schema
slapd.conf include /etc/openldap/schema/inetorgperson.schema
include /etc/openldap/schema/nis.schema
include /etc/openldap/schema/dyngroup.schema
pidfile /var/run/slapd.pid
argsfile /var/run/slapd.args
modulepath /usr/libexec/openldap
```

```
moduleload dds.la
moduleload dynlist.la
access to dn.base="" by * read
access to dn.base="cn=Subschema" by * read
access to *
    by self write
    by users read
    by anonymous auth
```

```
database config
rootdn cn=config
rootpw geheim
```

```
database hdb
suffix "o=avci,c=de"
rootdn "cn=admin,o=avci,c=de"
rootpw strengGeheim
checkpoint 1024 5
cachesize 2000
idl cachesize 6000
directory /var/lib/openldap/ldap/
index objectClass eq
index default pres,eq
index mail,telephoneNumber
index cn,sn,uid eq,sub
index entryUUID eq
access to attrs=entryTtl
    by dnattr=creatorsName manage
    by * read
```

<Weitere Access-Regeln>

```
overlay dynlist
dynlist-attrset groupOfURLs memberURL
```

```
overlay dds
dds-default-ttl 7200
dds-interval 1800
```

```
database monitor
```

Eine minimale LDIF-Datei zur Einrichtung eines Subtrees und einer Gruppe zeigt das folgende Listing. Die dynamischen Einträge werden dann im Subtree `ou=myHosts,o=avci,c=de` angelegt, während die Suchbasis die Gruppe `cn=myDynamicHosts,o=avci,c=de` ist.

Listing 17.3 *dynamic.ldif*

```
dn: ou=myHosts,o=avci,c=de
objectclass: organizationalUnit
ou: myHosts
description: subtree for dynamic ipHostNumbers
```

```
dn: cn=myDynamicHosts,o=avci,c=de
objectClass: groupOfURLs
cn: myDynamicHosts
memberURL: ldap:///ou=myHosts,o=avci,c=de \
?ipHostNumber?one?(objectclass=ipHost)
```

Bei der Erzeugung des `dynamicGroup`-Eintrages werden auch die operationalen Attribute `entryTtl` und `entryExpireTimestamp` erstellt, wobei der Wert für `entryExpireTimestamp` sich aus der Addition der Werte von `createTimestamp` oder `modifyTimestamp` und `entryTtl` ergibt. Nach Erreichen des Zeitstempels für `entryExpireTimestamp` wird der Eintrag gelöscht. Die Lebensdauer des Eintrages kann aber durch eine *refresh*-Operation verlängert werden. Abweichend von RFC 2589 wird bei OpenLDAP der `manageDIT-Control-OID` in Verbindung mit einem `Modify` des operationalen Attributes `entryTtl` als *Refresh*-Operation genutzt. Im Listing 17.2 auf Seite 176 wird als *Access-Regel* die Berechtigung *manage* angegeben. Diese Berechtigung führt dazu, dass die in der *Wer*-Regel definierte Identität das *manageDIT-Control* ausführen darf. Der Attributtyp `entryTtl` ist als *NO-USER-MODIFICATION* deklariert, der Wert darf also nicht verändert werden. Um nun als *Refresh*-Operation trotzdem den Zeitstempel zu modifizieren, wird die Berechtigung *manage* deklariert.

manageDIT Control ist ersetzt worden durch relax Control.

Das Tool `ldapmodify` wird mit dem zusätzlichen Parameter `-e relax` aufgerufen, um das Control zu aktivieren.

Mit dem Tool `web2ldap`, das in Abschnitt 18.3.1 auf Seite 187 vorgestellt wird, lässt sich die *Refresh*-Operation leicht bewerkstelligen: im Login-Fenster auf den Button *Connection Info* drücken, dann unter LDAP Options *Manage DIT* aktivieren, anschließend den infrage kommenden Eintrag aufrufen und den Wert für `entryTtl` modifizieren.

17.2 Konsolidierte Attributdarstellung

Eine Mischform aus *Proxy* und *Referral* bietet das Overlay *translucent*. Es können damit die Datensätze von zwei oder mehr Verzeichnisdiensten so zusammengefügt werden, dass sie eine konsolidierte, vereinigte Sicht bieten. Nehmen wir an, es existieren zwei Verzeichnisdienste, die beide ein Adressbuch führen, aber mit unterschiedlichem Inhalt. Statt

nun beide Server nach einer bestimmten Adresse zu durchsuchen, genügt es, einen Suchauftrag an einen Masterserver zu übergeben, der dann, durch das translucent Overlay, ein oder mehrere konsolidierte Suchergebnisse liefert. Als *Proof of Concept* stellen wir eine Basiskonfiguration vor.

```
include <die relevanten Schemadateien>
modulepath /opt/openldap/libexec/openldap/
moduleload translucent.la
...
database hdb
suffix o=avci,c=de
rootdn cn=admin,o=avci,c=de
rootpw geheim
access to *
    by cn=replicator,o=avci,c=de wx
    by users rcx
    by * auth x
index objectclass eq
index cn,sn,uid pres,eq,sub

overlay translucent
translucent_no_glue
uri ldapi://%2Fvar%2Frun%2Fldap
acl-bind binddn=cn=replicator,o=avci,c=de
    credentials=geheim
lastmod off
database monitor
```

Listing 17.4

slapd.conf
Masterserver

Die nachstehende LDIF-Datei enthält nur vier Einträge, die aber vollkommen ausreichend sind.

```
dn: o=avci,c=de
objectclass: organization
o: avci

dn: ou=adressbuch,o=avci,c=de
objectclass: organizationalUnit
ou: adressbuch

dn: cn=replicator,o=avci,c=de
objectclass: person
objectclass: uidObject
cn: replicator
sn: replicator
uid: replicator
userpassword: geheim
```

```
dn: cn=Franz Meier,ou=adressbuch,o=avci,c=de
objectclass: inetorgperson
cn: Franz Meier
sn: Meier
userPassword: strengGeheim
mail: fm@avci.de
telephonenumber: +4930543678
```

Eine Suche, wie im folgenden Beispiel demonstriert,

```
ldapsearch -Y GSSAPI -H ldap://localhost:9004 \
  -b ou=adressbuch,o=avci,c=de -s one \
  "(&(objectclass=inetorgperson)(sn=klu*))" \
  mail telephonenumber
```

liefert nun den gewünschten Eintrag, der sich auf dem Host localhost:389 befindet, zurück. Modifikationen an beiden Databases, sowohl der entfernten als auch der lokalen, sind prinzipiell möglich. Die entsprechenden Konfigurationsanleitungen befinden sich in der Manual Page `slapo-translucent(5)`.

17.3 Passwort-Regeln und Kontrolle

Die Informationstechnologie kreiert ständig neue Schlagwörter. Zur Zeit ist *Compliance* in aller Munde. Darunter ist die Übereinstimmung mit einem festen Regelwerk zu verstehen. Das Regelwerk setzt sich aus unternehmensinternen Richtlinien und externen Vorschriften und Gesetzen zusammen, die einerseits den sicheren Betrieb einer komplexen IT-Infrastruktur gewährleisten und andererseits die Kontrolle über die IT-Prozesse ermöglichen.

Ein Regelsatz zur Erstellung und Nutzung von Passwörtern ist zwingender Bestandteil aller Richtlinien. Solch ein Regelsatz wird als *Password Policy* bezeichnet. OpenLDAP ermöglicht die Umsetzung einer Password Policy durch das Overlay *ppolicy*. Die Basis dieses Overlays ist der IETF-Vorschlag *draft-behera-ldap-password-policy-07.txt*. Diesem Vorschlag liegt auch das Schema *ppolicy.schema* zugrunde, das natürlich inkludiert werden muss, sofern die Passwortkontrolle realisiert werden soll. Dieser Vorschlag und damit das Schema enthalten zwei Attribute, die besonderes Augenmerk verlangen.

Der Attributtyp *pwdAttribute* hat die Syntax OID, was bedeutet, dass der Wert dieses Attributtyps als OID notiert werden muss. Da das Overlay *ppolicy* zur Zeit nur den Attributtyp *userPassword* unterstützt, muss der OID 2.5.4.35 angegeben werden.

Der Attributtyp *pwdPolicySubentry* wird als *operational* mit *NO-USER-MODIFICATION* deklariert. Es ist also nicht ohne Weiteres möglich, einen Eintrag anzulegen, der dieses Attribut enthält, wie in Listing 17.7 auf der nächsten Seite dargestellt wird. Um eine solche LDIF-Datei zu laden, muss die *manageDIT-Control-Operation* aktiviert werden.

Falls das Tool *ldapadd* benutzt wird, ist zusätzlich der Schalter *-M* zu verwenden, die Manual Page *ldapadd(1)* gibt weitere Informationen, die Manual Page *slapo-ppolicy(5)* beschreibt die Konfiguration des Overlays und enthält eine vollständige Dokumentation der Attribute und Objektklassen. Das folgende Listing 17.5 zeigt die grundlegende Konfiguration.

```
include /etc/openldap/schema/ppolicy.schema
moduleload ppolicy.la
database hdb
suffix "o=avci,c=de"
rootdn "cn=admin,o=avci,c=de"
rootpw geheim
...
overlay ppolicy
ppolicy_default "cn=guests,ou=policies,o=avci,c=de"
ppolicy_use_lockout
```

Listing 17.5
Password-Policy-
Konfiguration

Das Regelwerk selbst wird in einem Eintrag als Attribut notiert. Hierbei kann ein globales Regelwerk in einem oder mehreren Einträgen oder direkt in einem User-Eintrag definiert werden. Falls viele Anwender vorhanden sind, ist die Beschreibung des Regelwerkes in einem gesonderten Eintrag vorzuziehen, wie in Listing 17.6 beschrieben wird. Hierbei ist zu berücksichtigen, dass alle Zeitangaben in *Sekunden* notiert werden müssen.

```
dn: ou=policies,o=avci,c=de
objectclass: organizationalUnit
ou: policies

dn: cn=anwender,ou=policies,o=avci,c=de
cn: anwender
objectClass: organizationalRole
objectClass: pwdPolicy
pwdAllowUserChange: TRUE
pwdAttribute: 2.5.4.35
pwdCheckQuality: 1
pwdExpireWarning: 86400
pwdGraceAuthNLimit: 2
pwdInHistory: 6
```

Listing 17.6
Anwender-Policy

```

pwdLockout: TRUE
pwdLockoutDuration: 1800
pwdMaxAge: 250000
pwdMaxFailure: 3
pwdMinAge: 3600
pwdMinLength: 6
pwdMustChange: TRUE
pwdSafeModify: FALSE

```

Ein User-Eintrag kann nur nach der Aktivierung der Passwort-Kontrolle angelegt werden.

Es ist wichtig zu bedenken, dass Anwender-Einträge erst nach der Aktivierung der Passwort-Kontrolle angelegt werden können. Die nachträgliche Erweiterung eines bereits bestehenden User-Eintrags ist nicht möglich. Wie im Listing 17.7 dargestellt, muss ein Anwender zusätzlich der auxiliären Objektklasse *pwdPolicy* angehören, das Attribut *pwdAttribute* wird von dieser Objektklasse als zwingend (MUST) vorgeschrieben. Sofern nur eine Password Policy vorliegt, kann diese über den Eintrag in *slapd.conf* mit dem Parameter *ppolicy_default* gesteuert werden und das operationale Attribut *pwdPolicySubentry* kann entfallen.

Listing 17.7
User-Eintrag

```

dn: cn=Hans Mustermann,ou=Mitarbeiter,o=avci,c=de
cn: Hans Mustermann
displayName: Mustermann
mail: hamu@avci.de
objectClass: inetOrgPerson
objectClass: pwdPolicy
postalAddress: 12345$Berlin$Werkstr. 123
postalCode: 12345
pwdAttribute: 2.5.4.35
sn: Mustermann
street: Werkstr. 123
telephoneNumber: +49.30.9876543
userPassword: geheim
pwdPolicySubentry: cn=anwender,ou=policies,o=avci,c=de

```

Die regelmäßige Änderung des Passwortes wird sinnvollerweise unter Verwendung des *Password Modify Extended Operation Control*, beschrieben in RFC 3062, durchgeführt. Jedes aktuelle Administrationstool ist in der Lage, diese Control-Operation auszuführen.