28.3 Samba verwaltet selbst Konten im LDAP

Keine externen Tools mehr notwendig

Seit Samba 3.0.25 kann Samba ohne externe Hilfsmittel Benutzer- und Maschinenkonten im LDAP anlegen und auch wieder entfernen. Dieses Setup setzt voraus, dass auf dem Domänencontroller der `winbindd`-Dämon läuft[2].

Die Samba-Konfiguration wird um einige Parameter erweitert. Zusätzlich zur in Kapitel 28 beschriebenen Konfiguration für Domänencontroller werden weitere Parameter notwendig. Zunächst wird Samba an das LDAP-Verzeichnis angebunden:

```
passdb backend = ldapsam
ldap admin dn = cn=samba,dc=samba,dc=org
ldap suffix = dc=samba,dc=org
ldap user suffix = ou=users
ldap group suffix = ou=groups
ldap machine suffix = ou=computers
ldap idmap suffix = ou=idmaps
ldap ssl = no
```

Diese Parameter bedeuten im Einzelnen:

- **passdb backend:** Hiermit geben wir an, in welchem Backend die Samba-Konten abgelegt werden. `ldapsam` zeigt auf einen LDAP-Server. Wird keine URL angegeben, wird der LDAP-Server lokal auf der Maschine erwartet.
- **ldap admin dn:** Diesen Benutzer verwendet Samba, um sich mit dem LDAP-Server zu verbinden. Das zugehörige Passwort muss noch hinterlegt werden. Hier sollte *nicht* der *rootdn* der LDAP-Datenbank angegeben werden[3]**!**
- **ldap suffix:** Base des LDAP-Verzeichnisses. Bei OpenLDAP wird das Suffix mit dem Parameter `suffix` in der Konfigurationsdatei `slapd.conf` festgelegt.
- **ldap user suffix:** LDAP-Container für Benutzerkonten
- **ldap group suffix:** LDAP-Container für Gruppen
- **ldap machine suffix:** LDAP-Container für Maschinenkonten
- **ldap idmap suffix:** LDAP-Container für das Identity Mapping (siehe Kapitel 29.4.3)
- **ldap ssl:** Dieser Parameter legt fest, ob die Daten zum LDAP-Server verschlüsselt werden sollen oder nicht. Je nach Setup Ihres LDAP-

[2] Früher galt, dass der Winbind nur auf Domänenmitgliedern benötigt wird. Das gilt inzwischen nicht mehr uneingeschränkt.

[3] Der *rootdn* kann unabhängig aller Zugriffsbeschränkungen immer alle Attribute lesen und vor allem auch verändern!

Servers muss dieser Wert angepasst werden. Mehr dazu erfahren Sie in Abschnitt 32.6.

Damit Samba überhaupt eine Verbindung zum LDAP-Server aufbauen kann, hinterlegt man noch das Passwort für das Konto, das mit dem Parameter `ldap admin dn` festgelegt wurde:

```
root@host:~ > smbpasswd -W
Setting stored password for "cn=samba,dc=samba,dc=org" in secrets.tdb
New SMB password:
Retype new SMB password:
root@host:~ >
```

Die obigen Parameter sorgen dafür, dass Samba die Benutzer aus dem LDAP beziehen kann. Um nun die LDAP-Daten selber verändern zu können, werden zusätzlich folgende Parameter gebraucht:

```
ldapsam:trusted = yes
ldapsam:editposix = yes
```

`ldapsam:trusted = yes` erlaubt Samba, direkt, d.h. ohne den *Name Service Switch*-Mechanismus (siehe Kapitel 29.4.1), auf das LDAP-Verzeichnis zuzugreifen. Dieser Parameter muss gesetzt sein, damit der eigentliche Parameter zum Editieren der LDAP-Daten, `ldapsam:editposix = yes`, überhaupt verwendet werden kann.

`ldapsam:trusted = yes` *Voraussetzung für* `ldapsam:editposix = yes`

Wenn Samba Benutzer anlegen soll, müssen irgendwie Benutzer-IDs verwaltet werden. Diese Aufgabe übernimmt der Winbind. Für den Winbind werden weitere Konfigurationsparameter[4] notwendig:

```
idmap uid = 10000-20000
idmap gid = 10000-20000
```

Erläuterung der Parameter:

- **idmap uid:** Bereich von Benutzer-IDs, für die der Winbind zuständig ist
- **idmap gid:** Bereich von Gruppen-IDs, für die der Winbind zuständig ist

Die Samba-Konfiguration ist nun so weit vorbereitet.

[4] Es sei an dieser Stelle erwähnt, dass diese Winbind-Konfiguration in Kombination mit dem editposix-Setup erst ab Samba 3.3.0 funktioniert. In Versionen zwischen 3.0.25 und 3.2.X werden weitere Parameter benötigt. In Samba 3.3 wurde die Konfiguration wieder vereinfacht.

28.3.1 LDAP-Datenbank bevölkern

Die LDAP-Datenbank ist zum jetzigen Zeitpunkt noch leer. Damit Samba Objekte anlegen kann, muss allerdings die Grundstruktur existieren. Für die erste Bevölkerung dieses Beispiel-Setups wurde folgende LDIF-Datei[5] benutzt:

```
dn: dc=samba,dc=org
objectClass: top
objectClass: dcObject
objectClass: organization
o: samba.org
dc: samba

dn: cn=samba,dc=samba,dc=org
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: samba
description: Samba user to connect to LDAP
userPassword: samba

dn: ou=users,dc=samba,dc=org
objectClass: top
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=samba,dc=org
objectClass: top
objectClass: organizationalUnit
ou: groups

dn: ou=idmap,dc=samba,dc=org
objectClass: top
objectClass: organizationalUnit
ou: idmap

dn: ou=computers,dc=samba,dc=org
objectClass: top
objectClass: organizationalUnit
ou: computers
```

[5] LDIF ist die Abkürzung für *LDAP Data Interchange Format*. Das ist ein ASCII-basierendes Dateiformat zur Darstellung von Informationen aus einem LDAP-Verzeichnis.

Diese LDIF-Datei wird mit folgendem Kommando importiert:

```
root@host:~ > ldapadd -x -D cn=Manager,dc=samba,dc=org⟵
-f samba.ldif -W
```

Jetzt sind zwar die Grundstruktur (Basis und Container für Benutzer, Maschinenkonten und Gruppen) und der `ldap admin dn` vorhanden, aber es fehlen noch einige Objekte, die Samba zwingend benötigt, wie z. B. das Gast-Konto *nobody* oder die Windows-Standardgruppen *Domänen Administratoren*, *Domänen Benutzer* und *Domänen Gäste*. Netterweise gibt es ein `net`-Unterkommando, das diese Objekte anlegt: `net sam provision`.

```
root@host:~ > net sam provision
Checking for Domain Users group.
Adding the Domain Users group.
Checking for Domain Admins group.
Adding the Domain Admins group.
Check for Administrator account.
Adding the Administrator user.
Checking for Guest user.
Adding the Guest user.
Checking Guest's group.
Adding the Domain Guests group.
```

Einen anschließenden Blick mit einem grafischen LDAP-Browser auf die LDAP-Datenbank sehen Sie in Abbildung 28-1.

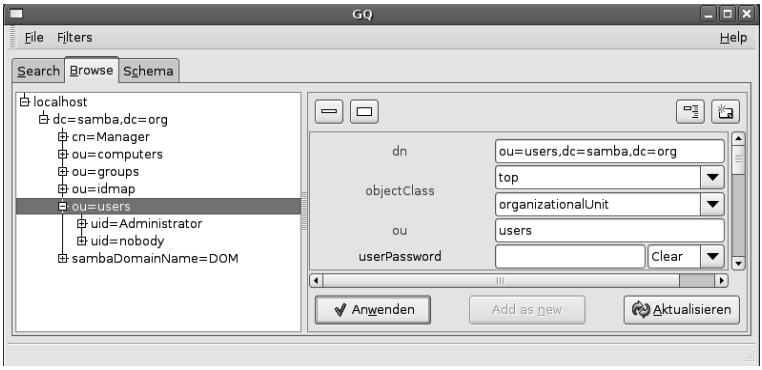

Abb. 28-1
LDAP-Datenbank

28.3.2 Systembenutzer aus dem LDAP beziehen

Wir haben nun im LDAP-Verzeichnis unter anderem einen Benutzer namens *Administrator* angelegt. Dieser Benutzer gehört der Objektklasse *posixAccount* an und ist somit ein Unix-Benutzer. Da-

mit das System die Benutzer nicht nur in der lokalen /etc/passwd-Datei, sondern auch im LDAP-Verzeichnis sucht, muss der nsswitch-Mechanismus entsprechend konfiguriert werden. Gleiches gilt auch für die Unix-Gruppen. Dazu wird das nsswitch-Modul *nss_ldap* in der Datei /etc/nsswitch.conf aktiviert:

```
passwd: compat ldap
group:  compat ldap
```

Voraussetzung hierfür ist, dass das nss_ldap-Modul auf dem System installiert ist. Weitere Informationen zum nsswitch-Mechanismus finden Sie in Kapitel 29.4.1.

Ob der Benutzer *Administrator* nun auch wirklich als Unix-Benutzer zur Verfügung steht, können wir z. B. mit dem Kommando id testen:

```
root@host:~ > id Administrator
uid=10000(Administrator) gid=10001(domadmins) groups=10001(domadmins)
```

nscd funkt oft dazwischen.

Es kann passieren, dass der *Name-Service-Cache-Dämon* (nscd) dafür sorgt, dass die Benutzerliste zwischengespeichert wird und aus Performance-Gründen nicht bei jedem Aufruf neu angefordert wird. So kann es vorkommen, dass der Benutzer nicht aufgelistet wird, obwohl alles richtig konfiguriert ist. Daher ist es empfehlenswert, den nscd für diese Tests zu stoppen (z. B. mit killall nscd).

28.3.3 Neuen Benutzer mit Samba anlegen

Der Clou ist jetzt, dass mit Samba-Boardmitteln neue Benutzer inklusive der Posix-Attribute angelegt werden können. Früher musste man Samba externe Skripte an die Hand geben, die die Unix-Benutzer anlegten. Das gehört nun endlich der Vergangenheit an!

Nicht alle Samba-Kommandos können die LDAP-Datenbank manipulieren. Lediglich Kommandos, die mit RPC-Calls arbeiten, legen auch Unix-Benutzer an. Das Kommando smbpasswd -a beispielsweise kann nur einem bereits existierenden Unix-Konto Samba-Attribute hinzufügen, aber nicht den Unix-Benutzer selbst anlegen. Das Kommando net rpc user add hingegen erledigt beide Aufgaben, setzt allerdings auch voraus, dass smbd und winbindd laufen.

Bevor wir neue Benutzer anlegen, muss der Samba-Benutzer *root* existieren. Da der Unix-Benutzer *root* bereits in der Datei /etc/passwd existiert, könnte man denken, dass lediglich die Samba-Attribute mit smbpasswd -a root hinzugefügt werden müssen. Das führt allerdings zu einem Problem:

Schauen wir mal die `ldapsearch`-Suche nach dem Benutzer *root* nach dem Aufruf von `smbpasswd -a root` an:

```
root@host:~ > ldapsearch -x -D "cn=Manager,dc=samba,dc=org" -W
uid=root -b dc=samba,dc=org
# extended LDIF
#
# LDAPv3
# base <dc=samba,dc=org> with scope subtree
# filter: uid=root
# requesting: ALL
#

# root, users, samba.org
dn: uid=root,ou=users,dc=samba,dc=org
uid: root
sambaSID: S-1-5-21-3998989574-2967633591-2819359875-1001
displayName: root
sambaNTPassword: C2AE1FE6E648846352453E816F2AEB93
sambaPasswordHistory: 0000000000000000000000000000000000000000000000000000000000000000
 00000000
sambaPwdLastSet: 1238321017
sambaAcctFlags: [U          ]
objectClass: sambaSamAccount
objectClass: account

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Es fällt auf, dass der Benutzer nicht zur Objektklasse *posixAccount* gehört. Wir haben jetzt also die Situation, dass das Unix-Konto aus der Datei `/etc/passwd` kommt, das Samba-Konto aber aus dem LDAP-Verzeichnis.

Probiert man nun, eine Verbindung mit `smbclient` aufzubauen, kommt es zu einer Fehlermeldung:

```
root@host:~ > smbclient -L localhost -U root
Enter root's password:
session setup failed: NT_STATUS_INTERNAL_DB_CORRUPTION
```

Unix- und Samba-Konto müssen im gleichen Backend sein!

Es ist nicht vorgesehen, dass das Unix-Konto in einem anderen Backend abgelegt ist als das Samba-Konto.

Löschen wir den Samba-Benutzer *root* also aus der LDAP-DB und legen ihn anschließend mit folgender LDIF-Datei als Unix-Benutzer wieder an:

```
dn: uid=root,ou=users,dc=samba,dc=org
objectClass: account
userid: root
objectClass: posixAccount
cn: root
uidNumber: 0
gidNumber: 10001
homeDirectory: /root
```

Die hier angegebene *gidNumber* entspricht der Gruppe der *Domänen Administratoren*. Je nachdem, welcher Bereich für Gruppen-IDs verwendet wird, muss dieser Wert eventuell angepasst werden. Wichtig ist, dass wirklich eine Gruppe mit dieser ID vorhanden ist.

Danach werden die Samba-Attribute mit `smbpasswd -a root` hinzugefügt. Ein Verbindungsaufbau mittels `smbclient` klappt anschließend problemlos:

```
smbpasswd -a root
New SMB password:
Retype new SMB password:
Added user root.
root@host:~ > smbclient -L localhost -U root%geheim
Domain=[DOM] OS=[Unix] Server=[Samba 3.3.2]

        Sharename       Type      Comment
        ---------       ----      -------
        link            Disk
        tmp             Disk
        samba           Disk
        IPC$            IPC       IPC Service (Samba 3.3.2)
```

Der Unix-Benutzer *root* ist jetzt sowohl in der `/etc/passwd` als auch in der LDAP-Datenbank vorhanden. Das stört allerdings nicht weiter. Ihn aus der `/etc/passwd` zu entfernen ist keine gute Idee, weil dann eine lokale Anmeldung nicht möglich wäre, wenn der LDAP-Dienst mal nicht erreichbar sein sollte (z. B. nach einer fehlerhaften Konfigurationsänderung oder einem Update).

Jetzt kann endlich ein neuer Testbenutzer angelegt werden:

```
root@host:~ > net rpc user add idefix pwidefix -U root
Enter root's password:
Added user 'idefix'.
```

Der Benutzer *idefix* wurde mit dem Passwort *pwidefix* angelegt. Vorsichtshalber testen wir noch, ob sowohl Unix- als auch Samba-Konto wirklich vorhanden sind:

```
root@host:~ > id idefix
uid=10001(idefix) gid=10000(domusers) groups=10000(domusers)
root@host:~ > pdbedit -L idefix
idefix:10001:idefix
```

Auch der Zugriff mit `smbclient` funktioniert einwandfrei:

```
root@host:~ > smbclient -L localhost -U idefix%pwidefix
Domain=[DOM] OS=[Unix] Server=[Samba 3.3.2]

        Sharename       Type      Comment
        ---------       ----      -------
        link            Disk
        tmp             Disk
        samba           Disk
        IPC$            IPC       IPC Service (Samba 3.3.2)
```

Da sich im Hinblick auf das Identity Mapping, beziehungsweise die Winbind-Konfiguration in den letzten Samba-Versionen immer wieder Kleinigkeiten geändert haben (und auch wahrscheinlich noch einmal ändern werden), müssen die Winbind-spezifischen Parameter eventuell je nach eingesetzter Version leicht angepasst werden. Getestet wurde die hier beschriebene Konfiguration mit Samba 3.3.4.

28.4 Homeverzeichnisse

Eine häufige Aufgabe eines PDC ist das Freigeben der Homeverzeichnisse der Benutzer. Natürlich könnte man jedes Homeverzeichnis explizit als Freigabe deklarieren und so den Benutzern zur Verfügung stellen, allerdings wäre das bei mehreren hundert Benutzern sehr aufwändig und damit unrealistisch. Aus diesem Grund bietet Samba die spezielle Freigabe `[homes]` an. Im einfachsten Falle sieht die Freigabedefinition so aus:

```
[homes]
valid users = %S
writeable = yes
```