

# Samba 4 als Active Directory

[https://wiki.ubuntuusers.de/Archiv/Howto/Samba4-Server\\_als\\_Active-Directory\\_Domain-Controller/](https://wiki.ubuntuusers.de/Archiv/Howto/Samba4-Server_als_Active-Directory_Domain-Controller/)

## Buch-Empfehlung

[https://www.rheinwerk-verlag.de/samba-4\\_4186/](https://www.rheinwerk-verlag.de/samba-4_4186/)

Achtung: 2. Auflage ist massiv überarbeitet (blaue Ausgabe)

## Samba-Tools

Tool	Beschreibung
testparm	Testen der smb.conf
smbcontrol	Steuert den 'Samba-Dienst
samba-tool	
smbstatus	Status von Verbindungen und Dateien anzeigen
net	
smbpasswd	Passwort eines Samba-Benutzers setzen
pdbedit	Samba-Datenbanken einsehen und editieren. VORSICHT!!
smbcalcs	
smbclient	Samba-Client, um sich auf einem SMB-Server zu verbinden
mount.cifs	
nmblookup	
tbdedit	
wbinfo	
rpcclient	

## Installationscript

Die folgende Dateien können verwendet werden, um eine rudimentäre Konfiguration vorzunehmen.

[install\\_samba.py](#)

```
#!/usr/bin/python3.5
# -*- coding: utf-8 -*-
"""
Created on Wed Jun 29 19:34:29 2016

@author: Richard Wegers
"""
import os
```

```
smb_conf= (
    "[global]",
    "workgroup = training",
    "netbios name = tsam-fs01",
    "server string = Training CIFS Server",
    "server role = standalone server",
    "[Public]",
    "path = /srv/samba/Public/",
    "comment = Dataiablage",
    "browseable = yes",
    "writeable = yes",
    "guest ok = yes"
)

# Hilfsfunktion, um Texte an eine Datei anzuhängen
# file: Pfad der Datei
# in_string: Inhalt, der angehängt werden soll als Dictionary
def write2file(file, in_string, mode):
    mytxt=""
    with open(file, mode) as myfile:
        for i in in_string:
            mytxt=mytxt + str(i) + "\n"
        print(mytxt, file=myfile)

os.system ("sudo -s")
os.system ("apt-get install -y samba winbind libnss-winbind smbclient
heimdal-clients ldb-tools")
# winbind zum Beitreten einer Domain
# libnss User Authorisierung
# smbclient SMB-Client für Freigaben
# heimdal Kerberos Implementierung
os.system ("mkdir -p /srv/samba/Public")
os.system ("chmod 1777 /srv/samba/Public")
os.system ("mv /etc/samba/smb.conf /etc/samba/smb.conf.org")
write2file ("/etc/samba/smb.conf", "smb_conf", "w")
os.system ("testparm /etc/samba/smb.conf")
os.system ("systemctl restart samba-ad-dc smbd nmbd winbind")
os.system ("systemctl status samba-ad-dc smbd nmbd winbind")
```

Im vorangegangenen Script wird eine erste smb.conf abgelegt. Die folgende Datei ist etwas ausführlicher.

### smb.conf

```
[global]
```

```
workgroup = training
netbios name = tsam-fs01
server string = Training CIFS Server
server role = standalone server

[Public]

path = /srv/samba/Public/
comment = Dateifreigabe
browseable = yes
writeable = yes
guest ok = yes

[homes]
comment = Heimatverzeichnis
browseable = yes
guest ok = yes
writeable = yes
valid users = %S
# valid users = %S nur Eigentümer $S darf sich auf ein Home-
Ordner verbinden
```

Geänderte Konfiguration prüfen und dann neu einladen:

```
testparm /etc/samba/smb.conf
smbcontrol all reload-config
```

## Aufgabe 2: Arbeiten mit smbclient

User test1 anlegen und home-Ordner (-m):

```
useradd -m test1
```

Linux-Passwort für test1 setzen:

```
passwd test1
```

Solange der Nutzer nicht als Samba-Nutzer angelegt wurde, kann er mit `pdbedit -L` nicht angezeigt werden.

Samba-Passwort für test1 setzen:

```
smbpasswd test1
```

Verbinden mit eines Users test1 auf den home-Ordner eines Samba-Servers:

```
smbclient -U test1 //tsam-fs01/homes
```

Hochladen der lokalen Datei testfile in den Ordner testdir mit neuem Dateinamen newfile:

```
put testfile testdir\newfile
```

Status eines Users training auf dem Samba-Server tsam-fs01 anzeigen:

```
smbclient -L //tsam-fs01 -U training
```

Listen aller Samba-Nutzer:

```
pdbedit -L
```

Listen des Nutzers test1 Samba-Nutzer:

```
pdbedit -u test1 -v
```

## Aufgabe 3: Gruppenlaufwerke

Erst wird eine Gruppe gruppe1 angelegt und im Anschluss die beiden Nutzer training und test1 in die Gruppe aufgenommen. getent Get entities, um die Gesamtsicht auf alle Benutzer zu bekommen. less /etc/groups würde nur die lokalen Benutzer anzeigen, nicht aber die aus z.B. LDAP-Servern.

```
groupadd gruppe1
usermod -a -G gruppe1 training
usermod -a -G gruppe1 test1
getent passwd
getent group
mkdir /srv/samba/Gruppe1/
chown root.gruppe1 /srv/samba/Gruppe1/
chmod 770 /srv/samba/Gruppe1/
ll /srv/samba
```

```
[Gruppe1]
    path = /srv/samba/Gruppe1/
    comment = Gruppe 1 Austauschordner
    browseable = yes
    writeable = yes
    valid users = @gruppe1
    guest ok = no
# erst im zweiten Schritt hinzufügen
    force create mode = 0660
    force directory mode = 0770
    force group = gruppe1
```

Die force Parameter werden als UND-Verknüpfung auf die bestehenden Rechte gelegt. Damit kann also für die Ordner (0770) und Dateien (0660) die Gruppe gruppe1 jeweils mitberechtigt werden.

hide unreadable zeigt nur noch die Freigaben an, die der Benutzer überhaupt nutzen darf. Z.B. kann im /homes/-Ordner nur seinen eigenen Home-Ordner sehen und alle anderen nicht.

## Aufgabe 4 Active Directory

Die folgenden Befehle sind auf dem DC auszuführen.

```
apt-get install samba winbind libnss-winbind smbclient heimdal-clients ldb-tools
systemctl stop smbd nmbd winbind samba-ad-dc
mv /etc/samba/smb.conf /etc/samba/smb.conf.org
samba-tool domain provision
```

samba-tool domain provision macht aus dem Server eine DC.

Antworten für samba-tool:

```
Realm: TRAINING.EXAMPLE
Domain: TRAINING (Default)
Rolle: ENTER
DNS-Backend: ENTER
DNS-Forwarder: 192.168.122.1 (bzw. localhost)
```

**HINWEIS:** Domainname sollte nicht mehr auf local enden, da dies anderweitig zukünftig benutzt wird.

Alle Clients müssen den DC als DNS eingetragen haben. Im DC kann man einen sogenannten DNS-Forwarder wird der übergeordnete (z.B. Default-Router oder Provider-DNS) eingetragen werden.

Ausgabe von samba-tool am Ende der Installation:

```
Server Role:      active directory domain controller
Hostname:         tsam-dc01
NetBIOS Domain:  TRAINING
DNS Domain:       training.example
DOMAIN SID:       S-1-5-21-779612574-3552205093-1178054674
```

Inhalt smb.conf

```
# Global parameters
[global]
    workgroup = TRAINING
    realm = TRAINING.EXAMPLE
    netbios name = TSAM-DC01
    server role = active directory domain controller
    dns forwarder = 192.168.122.1 # Virtualbox Default-DNS bei NAT-Netzen
```

```
[netlogon]
  path = /var/lib/samba/sysvol/training.example/scripts
  read only = No

[sysvol]
  path = /var/lib/samba/sysvol
  read only = No
```

Überprüfung, welche Dienste aktiv sind mit `netstat -tunlp | egrep sa?mb | less`

netstat-Parameter:

```
t -> tcp
u -> udp
n -> numeric
l -> listen
p -> process-Id
```

egrep sa?mb:

```
'sa' wobei wegen des '?' das 'a' da sein kann, aber nicht muss.
Filter: 'smb' und 'samb'
```

Ports (???? unvollständig) s.S.8 Active Directory:

```
135
389, 636 LDAP
CIFS
Kerberos
NetBIOS
```

`systemctl status samba-ad-dc - samba` → Samba 4.0 - smb → Samba 3

Die Verwaltung der DC findet unter folgendem Verzeichnis statt: `/var/lib/samba/sysvol` Im Ordner `training.example` findet sich die Domain wieder. Unter `Policies` werden die GPOs als Dateien abgelegt.

`samba-tool gpo listall` anzeigen aller GPOs

Vollständiger Ordner zu den GPOs:

```
/var/lib/samba/sysvol/training.example/Policies/
```

**Wichtig:** DC sollte **NICHT** als Fileserver genutzt werden. Ein DC hat andere Anforderungen an Backup/Wiederherstellung als eine Fileserver.

# Aufgabe 5 Domain-Mitglied

Siehe Vorgehensweise auf Arbeitsblatt 8.

Zusätzliche Nutzer über den Windows-Cleint anlegen. WindowsTH-RSAT\_TP5-x64

Verwaltung → Active Directory Benutzer und Gruppen

Hier die zusätzlichen Nutzer und Gruppen anlegen.

Aus dem DC mit `watch -n 1 smbstatus` den aktuellen Status jeweils nach einer Sekunde aktualisiert anzeigen (`-n 1`).

## Protokoll-Analyse des Domain-Beitritt

Wie erhält der Client den Namen des DCs?

DNS-Anfrage auf `_ldap._tcp.dc._msdcs.training.example` als SRV-Record (service-Record). Dies ist eine fixe Konvention für einen Microsoft-basierten LDAP-Server. Als Antwort erhält der Client den `tsam-dc01.training.example` als Namen des DC.

Im Anschluss findet die Anmeldung des Administrator-Benutzers in Form einer Kerberos-Anfrage (Protocol: KRB5).

Im ADSI-Editor kann als AD-Explorer verwendet werden.

## Best Practice für AD DC

DC sollten mehrfach vorhanden sein, damit eine Ausfallsicherheit gegeben ist.

Funktional Level kann alte Betriebssysteme ausschließen.

DC nicht zurücksetzen. Ansonsten kommt es zwangsläufig zu Synchronisationsproblemen. KEINE Snapshots machen. Besser mehrere DC laufen lassen, damit immer einer noch aktiv ist.

Vorgehensweise:

- ein funktioniert noch einwandfrei
- weitere DCs neu aufsetzen und in diese Domain aufnehmen und neu synchronisieren lassen.

sysvol-Replikation unter Windows automatisch. Unter Linux mit Samba muss dies handisch gemacht werden. vergleiche S.24 rsync-Script

## FSMO-Rollen

FSMO-Rollen regeln die Zuständigkeit für Schemata (Schema-Master), Infrastruktur (Phantom-Objekt-

Verwaltung) und aktuelle Passwörter (PDC). Kritisch ist hier der PDC. Dieser sollte möglichst zeitnah wieder gestartet werden. Ein Neustart ist hier unkritisch. Ein Restore aus einem älteren Backup ist kritisch, da hier Veränderungen seit dem Backup-Zeitpunkt nicht repliziert werden!

## Aufgabe 6: Linux als Mitglied im AD

Seite 43 im Skript.

sssd als Zugangsverfahren zu AD nutzen. Cached Passwörter für eine gewisse Zeit und kann eigenständig Anmeldungen durchführen.

winbind ist älter und kann nicht so viel wie sssd.

Die `globals`-Sektion muss um den folgenden Teile ergänzt werden.

```
[global]

workgroup = training
netbios name = tsam-fs01
server string = Training CIFS Server
server role = member server
realm = TRAINING.EXAMPLE
idmap config * : range = 10000 - 19999
idmap config TRAINING : backend = rid
idmap config TRAINING : range = 100000 - 199999
winbind enum users = yes
winbind enum groups = yes
```

Mit `idmap config *` wird ein Bereich für alle nicht-Domain-Gruppen/User, also z.B. die lokalen Gruppen/User vorgegeben. Mit den folgenden Konfigurationen:

- `idmap config TRAINING : backend = rid`
- `idmap config TRAINING : range = 100000 - 199999`

Werden die Objekte der Domain TRAINING konfiguriert. `backend = rid` verwendet für die UID die RID, also den letzten Teil der SID. Über den Bereich werden sie von den lokalen Gruppen/User getrennt.

## Gruppenlaufwerk für Klasse11b freigeben

Zunächst müssen die Ordner-Gruppe angepasst werden:

```
chgrp -R "TRAINING\klasse11b" /src/samba/Gruppe1/
```

Die Share-Freigabe für `gruppe1` muss in `/etc/samba/smb.conf` angepasst werden:

```
[Gruppe1]
    path = /srv/samba/Gruppe1
    comment = Gruppe 1 Austausch
    browseable = yes
    writeable = yes
    valid users = @TRAINING\klasse11b
    guest ok = no
    force create mode = 0660
    force directory mode = 0770
    force group = TRAINING\klasse11b
```

Danach samaba neustarten:

```
systemctl restart smbd
```

oder

```
smbcontrol all reload-config
```

Überprüfen lässt sich der Erfolg durch folgenden Befehl:

```
smbclient -U "TRAINING\schueler1" -L //tsam-fs01/Gruppe1
```

Die Ausgabe sieht dann wie folgt aus:

```
Enter TRAINING\schueler1's password:
Domain=[TRAINING] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

  Sharename      Type      Comment
  -----      -
  Public         Disk     Dateifreigabe
  homes         Disk     Heimatverzeichnis
  Gruppe1       Disk     Gruppe 1 Austausch
  IPC$          IPC      IPC Service (Training CIFS Server)
  schueler1     Disk     Heimatverzeichnis
Domain=[TRAINING] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

  Server          Comment
  -----
  Workgroup       Master
  -----
  WORKGROUP      TSAM-FS01
```

# Aufgabe 8 Benutzerverwaltung unter Linux

Mit samba-tool kann direkt auf dem DC gearbeitet werden. Hier hilft wie immer ein Eingabe des

Befehls, um sich alle möglichen Parameter anzeigen zu lassen.

Generelle Befehle mit denen man die Domain einrichten kann: **ACHTUNG:** Dies Befehle sollten nur zu Testzwecken verwendet werden, da sie die Sicherheit des Systems massiv herabsetzen!!

```
samba-tool domain passwordsettings show
samba-tool domain passwordsettings set --complexity off
samba-tool domain passwordsettings set --min-pwd-length 4
```

Passwörter dürfen jetzt trivial und sehr kurz (mind. 4 Zeichen) sein.

## Anlegen einer Gruppe und hinzufügen von Usern

```
samba-tool group list # Anzeigen aller Gruppen
samba-tool user list # Anzeigen aller User

samba-tool group add GRUPPENNAME
```

[add\\_grps2samba.sh](#)

```
#!/bin/bash
for g in Informatik Leitung Verwaltung
do
  samba-tool group add $g
done
```

Anlegen eines Nutzers hmeier mit dem Vornamen Carl und dem Nachnamen Schmidt.  
Anschließend wird das initiale Passwort des Benutzers erfragt.

```
samba-tool user add hmeier --given-name=Carl --surname=Schmidt
New Password:
Retype Password:
User 'hmeier' created successfully
```

Hinzufügen der Nutzer in die Gruppen:

```
samba-tool group addmembers Lehrer hmeier,gmeier,cschmidt
samba-tool group addmembers Informatik hmeier,gmeier
samba-tool group addmembers Verwaltung omueller
samba-tool group addmembers Leitung hmeier,cschmidt
```

Zur Überprüfung kann mit den folgenden Befehlen der Inhalt der einzelnen Gruppen angezeigt werden.

```
samba-tool group listmembers Lehrer
Ausgabe:
hmeier
```

```
cschmidt
gmeier
samba-tool group listmembers Leitung
samba-tool group listmembers Verwaltung
samba-tool group listmembers Informatik
```

## Aufgabe 10 LDAP-Queries / Filtern des LDAP-Datenbank

Um die Gruppenmitglieder einzelner Gruppen zu prüfen, kann folgender Befehl verwendet werden:

```
ldbsearch -H ldap://localhost/ -U "TRAINING\Administrator"
"(objectclass=group)" member
```

Ausgabe:

```
# record 35
dn: CN=Informatik,CN=Users,DC=training,DC=example
member: CN=Heinrich Meier,CN=Users,DC=training,DC=example
member: CN=Gerlinde Meier,CN=Users,DC=training,DC=example
```

Um die Zugehörigkeit eines Users zu diversen Gruppen anzuzeigen, kann der folgende Befehl verwendet werden:

<code>

```
ldbsearch -H ldap://localhost/ -U "TRAINING\Administrator"
"(objectclass=person)" memberOf
```

Ausgabe:

```
...
# record 8
dn: CN=Carl Schmidt,CN=Users,DC=training,DC=example
memberOf: CN=Lehrer,CN=Users,DC=training,DC=example
memberOf: CN=Leitung,CN=Users,DC=training,DC=example
```

Alternativ kann man auch direkt auf die Datei (/var/lib/samba/private/sam.ldb) auf dem DC gehen, falls der LDAP-Zugriff nicht mehr funktioniert. Hier ist keine Anmeldung per -U mehr nötig.

```
ldbsearch -H /var/lib/samba/private/sam.ldb "(objectclass=person)" memberOf
```

## Gezieltes Filtern von Objekten und Attributen

Um gezielt zu filtern kann man Ausdrücke klammern. Weiterhin wird die search-base auf den Bereich Users eingeschränkt (-b cn=Users,dc=training,dc=example):

```
ldbsearch -H ldap://localhost -U "/TRAINING\Administrator" -b
cn=Users,dc=training,dc=example
```

```
"(&(objectclass=group)(sAMAccountName=Leitung))" member
```

Ausgabe:

```
# record 1
```

```
dn: CN=Leitung,CN=Users,DC=training,DC=example
```

```
member: CN=Heinrich Meier,CN=Users,DC=training,DC=example
```

```
member: CN=Carl Schmidt,CN=Users,DC=training,DC=example
```

```
# returned 1 records
```

```
# 1 entries
```

```
# 0 referrals
```

## Aufgabe 11: LDAP-Objekte verändern

OFFEN: Konsistenz bedrohend, falls Änderungen nicht korrekt durchgeführt werden.

Nur über entsprechende Konsistenzbewahrende Tools durchführen. z.B. ldbrename

## Aufgabe 12: Wiederherstellung gelöschter Objekte (AD spezifisch)

[enablerecyclebin](#)

```
#!/usr/bin/env python
#
# enabled the Recycle Bin optional feature
#
import optparse
import sys

# Find right directory when running from source tree
sys.path.insert(0, "bin/python")

import samba
from samba import getopt as options, Ldb
from ldb import SCOPE_BASE
import sys
import ldb
from samba.auth import system_session

parser = optparse.OptionParser("enablerecyclebin <URL>")
sambaopts = options.SambaOptions(parser)
parser.add_option_group(sambaopts)
credopts = options.CredentialsOptions(parser)
```

```
parser.add_option_group(credopts)
parser.add_option_group(options.VersionOptions(parser))

opts, args = parser.parse_args()
opts.dump_all = True

if len(args) != 1:
    parser.print_usage()
    sys.exit(1)

url = args[0]

lp_ctx = sambaopts.get_loadparm()

creds = credopts.get_credentials(lp_ctx)
sam_ldb = Ldb(url, session_info=system_session(), credentials=creds,
lp=lp_ctx)

# get the rootDSE
res = sam_ldb.search(base="", expression="", scope=SCOPE_BASE,
attrs=["configurationNamingContext"])
rootDse = res[0]

configbase=rootDse["configurationNamingContext"]

# enable the feature
msg = ldb.Message()
msg.dn = ldb.Dn(sam_ldb, "")
msg["enableOptionalFeature"] = ldb.MessageElement(
    "CN=Partitions," + str(configbase) + ":766ddcd8-acd0-445e-f3b9-
a7f9b6744f2a",
    ldb.FLAG_MOD_ADD, "enableOptionalFeature")
res = sam_ldb.modify(msg)

print "Recycle Bin feature enabled"
```

## Aufgabe 14: Kerberos

Kerberos Folien

### Challenge-Response

#### NTLMv1

1. Passwort wird gehasht auf dem Server abgelegt

2. Server schickt ein Challenge an den Client
3. Client verschlüsselt die Challenge mit dem gehashten Passwort
4. Client schickt das verschlüsselte Challenge an den Server
5. Server ermittelt ebenfalls das verschlüsselte Challenge mittels des gespeicherten gehashten Passwort

Verfahren NTLMv1 gilt als unsicher.

## Ticket-System

1. User (Principal) meldet sich bei AS <sup>1)</sup> und bekommt ein TGT <sup>2)</sup>
2. User gibt das TGT an den TGS <sup>3)</sup> und erhält ein ST <sup>4)</sup>, welches mit dem Hash des entsprechenden Maschinen Konto des Service geschlüsselt wurde.
3. User kann nun mit dem ST an eigentlichen Dienst herantreten und sich dort anmelden.
4. Möchte der User einen zweiten Dienst nutzen, dann muss er für diesen ein neues ST beim TGS holen

## Linux-Konfiguration

```
cat /var/lib/samba/private/krb5.conf
[libdefaults]
    default_realm = TRAINING.EXAMPLE
    dns_lookup_realm = false
    dns_lookup_kdc = true
```

Ticket am Kerbos holen:

```
kinit Administrator@TRAINING.EXAMPLE
klist

Ausgabe:
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: Administrator@TRAINING.EXAMPLE

    Issued                Expires                Principal
Dec  8 13:53:27 2016  Dec  8 23:53:24 2016
krbtgt/TRAINING.EXAMPLE@TRAINING.EXAMPLE
```

Die letzte Zeile zeigt das TGT als krbtgt an.

Man kann pro User nur ein TGT erhalten. Will man sich mit unterschiedlichen Usernamen an einem AD anmelden, dann kann folgendes verwendet werden:

```
kinit -c testfile schueler1@TRAINING.EXAMPLE
klist -c testfile
```

```
Ausgabe:
Credentials cache: FILE:testfile
Principal: schueler1@TRAINING.EXAMPLE
```

Issued	Expires	Principal
Dec 8 13:55:02 2016	Dec 8 23:54:58 2016	
krbtgt/TRAINING.EXAMPLE@TRAINING.EXAMPLE		

Pro Anmeldung erhält man ein eigenes Ticket:

```
smbclient -L tsam-fs01 -k
```

```
Ausgabe:
Domain=[TRAINING] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
```

Sharename	Type	Comment
-----	----	-----
Public	Disk	Dateifreigabe
homes	Disk	Heimatverzeichnis
Gruppel	Disk	Gruppe 1 Austausch
IPC\$	IPC	IPC Service (Training CIFS Server)
administrator	Disk	Heimatverzeichnis

```
Domain=[TRAINING] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
```

Server	Comment
-----	-----
TSAM-FS01	Training CIFS Server

Workgroup	Master
-----	-----
TRAINING	TSAM-FS01

```
>> klist
```

```
Ausgabe:
Credentials cache: FILE:/tmp/krb5cc_0
Principal: Administrator@TRAINING.EXAMPLE
```

Issued	Expires	Principal
Dec 8 13:53:27 2016	Dec 8 23:53:24 2016	
krbtgt/TRAINING.EXAMPLE@TRAINING.EXAMPLE		
Dec 8 13:58:24 2016	Dec 8 23:53:24 2016	cifs/tsam-fs01@TRAINING.EXAMPLE

In der letzten Zeile ist nun ein neues Ticket für die CIFS-Freigabe

Unter Windows funktioniert der klist-Befehl ebenfalls. Ausgabe:

```
Aktuelle Anmelde-ID ist 0:0x2a443
```

```
Zwischengespeicherte Tickets: (2)
```

```
#0> Client: Administrator @ TRAINING.EXAMPLE
Server: krbtgt/TRAINING.EXAMPLE @ TRAINING.EXAMPLE
KerbTicket (Verschlüsselungstyp): AES-256-CTS-HMAC-SHA1-96
Ticketkennzeichen 0x40e00000 -> forwardable renewable initial
pre_authent
Startzeit: 12/8/2016 14:02:01 (lokal)
Endzeit: 12/9/2016 0:02:01 (lokal)
Erneuerungszeit: 12/15/2016 14:02:01 (lokal)
Sitzungsschlüsseltyp: AES-256-CTS-HMAC-SHA1-96
Cachekennzeichen: 0x1 -> PRIMARY
KDC aufgerufen: tsam-dc01.training.example

#1> Client: Administrator @ TRAINING.EXAMPLE
Server: cifs/tsam-fs01 @ TRAINING.EXAMPLE
KerbTicket (Verschlüsselungstyp): RSADSI RC4-HMAC(NT)
Ticketkennzeichen 0x40a80000 -> forwardable renewable pre_authent
0x80000
Startzeit: 12/8/2016 14:02:01 (lokal)
Endzeit: 12/9/2016 0:02:01 (lokal)
Erneuerungszeit: 12/15/2016 14:02:01 (lokal)
Sitzungsschlüsseltyp: RSADSI RC4-HMAC(NT)
Cachekennzeichen: 0
KDC aufgerufen: tsam-dc01.training.example
```

## Aufgabe 15: GPOs bearbeiten

```
samba-tool gpo listall
```

Ausgabe:

```
GPO : {31B2F340-016D-11D2-945F-00C04FB984F9}
display name : Default Domain Policy
path :
\\training.example\sysvol\training.example\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}
dn :
CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=training,DC=example
version : 0
flags : NONE

GPO : {6AC1786C-016F-11D2-945F-00C04FB984F9}
display name : Default Domain Controllers Policy
path :
\\training.example\sysvol\training.example\Policies\{6AC1786C-016F-11D2-945F-00C04FB984F9}
dn :
CN={6AC1786C-016F-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=training,DC=example
```

```

version      : 0
flags       : NONE

GPO         : {CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF}
display name : Schüler Policy
path        :
\\training.example\SysVol\training.example\Policies\{CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF}
dn          :
CN={CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF},CN=Policies,CN=System,DC=training,DC=example
version     : 589824
flags       : NONE

```

Man kann Geltungsbereich eines GPO-Containers gezielt ansehen. Hier: Alles unterhalb der OU=Schüler

```
samba-tool gpo listcontainers CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF
```

Ausgabe:

```

Container(s) using GPO CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF
  DN: OU=Schüler,OU=Schule,DC=training,DC=example

```

Für einzelne Accounts:

```
samba-tool gpo list schueler1
```

Ausgabe:

```

GPOs for user schueler1
  Schüler Policy {CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF}
  Default Domain Policy {31B2F340-016D-11D2-945F-00C04FB984F9}
samba-tool gpo list cschmidt

```

Ausgabe:

```

GPOs for user cschmidt
  Default Domain Policy {31B2F340-016D-11D2-945F-00C04FB984F9}

```

Beim Benutzer cschmidt kann man sehen, dass die zuvor angelegte GPO Schüler Policy nicht eingebunden ist.

Auf Verzeichnisebene kann man die Policies unter:

```
/var/lib/samba/sysvol/training.example/Policies/{CD2C83FD-4EEB-4DCD-88F9-97FE462C8CAF}
```

finden.

## Aufgabe 16: DNS im Active Directory

Im ADSI-Editor eine neue Verbindung herstellen über den Definierten Namen und Namenskontext . . mit folgendem Ziel:

```
DC=DomainDNSZones, DC=training,DC=example
```

## Aufgabe 17: ACLs

Access Control Lists verwalten.

Über die eigentlichen Filesystem-Zugriffsrechte hinaus weitere Berechtigungen verwalten. Z.B. Zugriff von verschiedenen Gruppen auf einen Ordner. Dies ist über die regulären Berechtigungen nicht möglich.

## Aufgabe 18: Roaming Profiles

## Aufgabe 19: Distributed File System (DFS)

## Aufgabe 20: Konfiguration per Registry

1)

Authentication Service

2)

Ticket Granting Ticket

3)

Ticket Granting Service

4)

Service Ticket

From:  
<http://www.kopfload.de/> - kopfload - Lad Dein Hirn auf!

Permanent link:  
[http://www.kopfload.de/doku.php?id=allgemein:howto:samba\\_ad&rev=1481269587](http://www.kopfload.de/doku.php?id=allgemein:howto:samba_ad&rev=1481269587)

Last update: 2025/11/19 16:13

