

**DezSys-Protokoll**

**DezSysLabor-04 "Authentifizierung & Autorisierung"**

**Dezentrale Systeme**

**5BHITT 2015/16**

**Thomas Stedronsky**

|  |  |
| --- | --- |
|  | **Version 1.0** |
| **Note:** | **Begonnen am 08. Jänner 2016** |
| **Betreuer: Th. Micheler** | **Beendet** |

Inhaltsverzeichnis

[1 Einführung 3](#_Toc440014080)

[1.1 Ziele 3](#_Toc440014081)

[1.2 Voraussetzungen 3](#_Toc440014082)

[1.3 Aufgabenstellung 3](#_Toc440014083)

[2 Ergebnisse 4](#_Toc440014084)

[2.1 LDAP Vorbereitungen 4](#_Toc440014085)

[2.2 Java Implementierung 4](#_Toc440014086)

[2.3 LDAP Search 5](#_Toc440014087)

[2.4 LDAP Modify 7](#_Toc440014088)

[3 Quellen 9](#_Toc440014089)

[4 Abbildungsverzeichnis 10](#_Toc440014090)

# Einführung

Diese Übung soll zur Vertiefung der Begriffe "Authentifizierung und Autorisierung" dienen.

## Ziele

Das Ziel dieser Übung ist die Funktionsweise eines Verzeichnisdienstes zu verstehen und Erfahrungen mit der Administration auszuprobieren. Ebenso soll die Verwendung des Dienstes aus einer Anwendung heraus mit Hilfe der JNDI geübt werden.

Authentifizierung bedeutet hier, dass per Username und Passwort eine Anmeldung beim Verzeichnisdienst erfolgt. Autorisierung wird hier im Zusammenhang mit Service-Gruppen und zugeordneten Usern durchgefuehrt.

## Voraussetzungen

* Grundlagen Verzeichnisdienst
* Administration eines LDAP Dienstes
* Verwendung von Commandline Werkzeugen fuer LDAP (LDAPSEARCH, LDAPMODIFY)
* Grundlagen der JNDI API für eine JAVA Implementierung
* Verwendung einer virtuellen Instanz für den Betrieb des Verzeichnisdienstes

## Aufgabenstellung

Mit Hilfe der zur Verfuegung gestellten VM wird ein vorkonfiguriertes LDAP Service zur Verfuegung gestellt. Dieser Verzeichnisdienst soll um folgende Eintraege erweitert werden. Das verwendete Namensschema (eg. group.service1 oder vorname.nachname) soll fuer alle Eintraege verwendet werden.

* 5 Posix Groups (beliebe Zuweisung von UserIDs)
* 10 User Accounts

Weiters soll eine Java-Applikationen zur Authentifizierung und Autorisierung entwickelt werden. Folgende Fragestellungen stehen dabei im Mittelpunkt:

* Sind Username und Passwort korrekt?   
  (Identifikation des Benutzers)
* Ist der User berechtigt ein bestimmtes Service zu nutzen?  
  (Benutzer-Berechtigung)

# Ergebnisse

## LDAP Vorbereitungen

Zu aller erst mussten die Gruppen und die User für die Übung angelegt werden. In den folgenden Bildern kann man die Maske für das Gruppen bzw. User anlegen sehen.

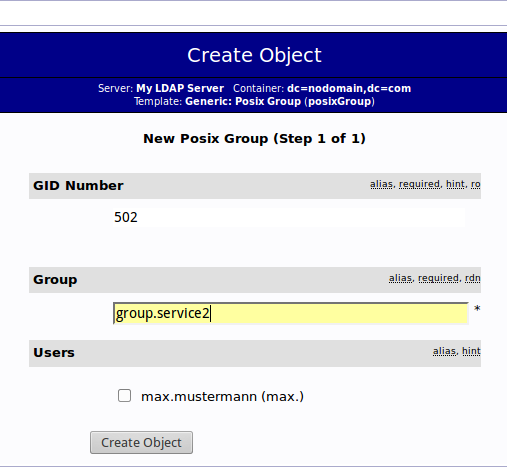


Abbildung 1 create\_group

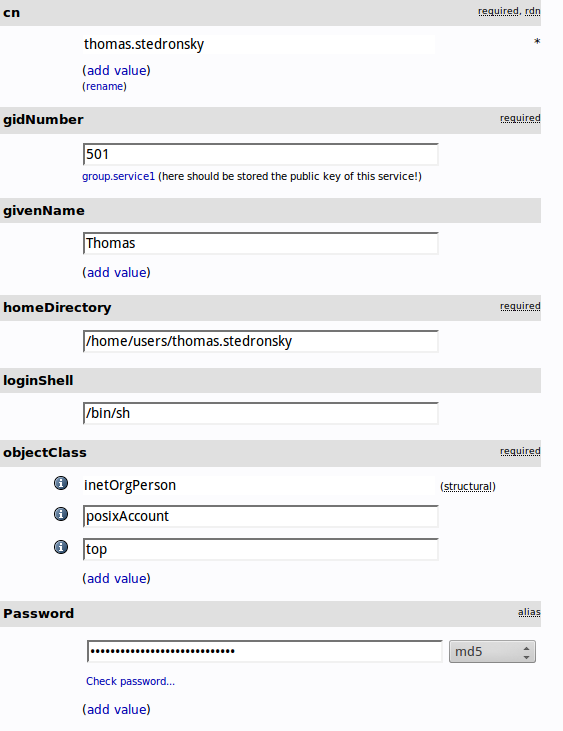


Abbildung 2 create\_user

## Java Implementierung

Nach dem die User angelegt worden sind, konnte mit der Java Implementierung begonnen werden. Dafür wurde ein Example von [1] verwendet.

## LDAP Search

-D … steht hierbei für den User mit dem auf das LDAP zugegriffen wird.

-b … ist der konkrete Suchbefehl

ldapsearch -h 127.0.0.1 -p 389 -D "cn=thomas.stedronsky,dc=nodomain,dc=com" -W -b "cn=group.service1,dc=nodomain,dc=com" memberUid

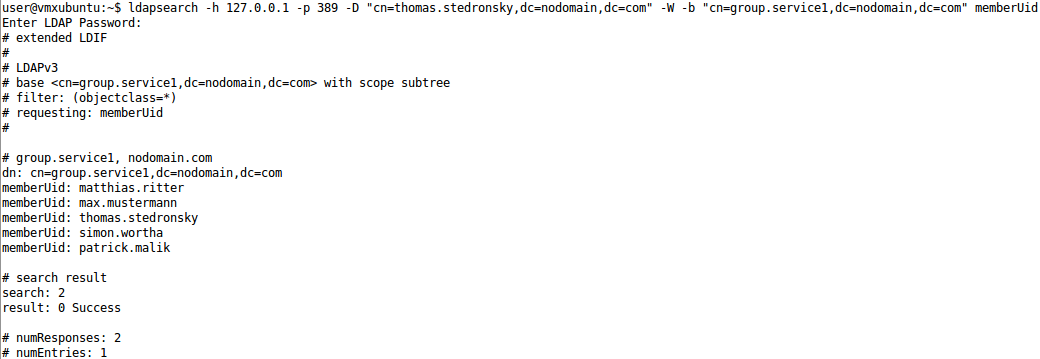


Abbildung 3 ldap\_search\_1

Bei diesem ldapsearch wird gezielt nach den Membern anhand der MemberUid der group.service1 gesucht.

ldapsearch -h 127.0.0.1 -p 389 -D "cn=thomas.stedronsky,dc=nodomain,dc=com" -W -b "cn=thomas.stedronsky,dc=nodomain,dc=com" UidNumber

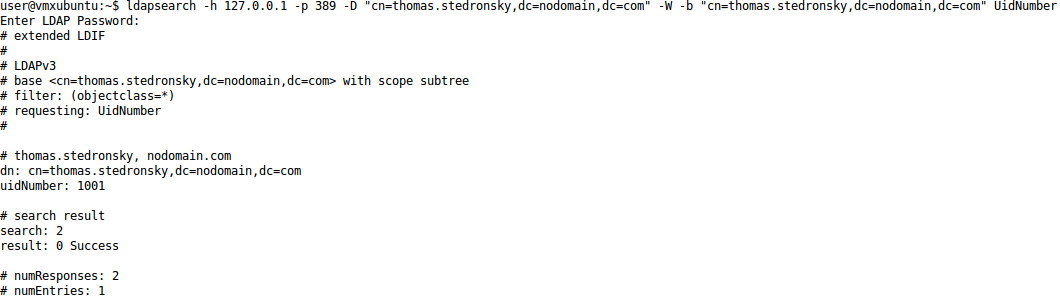


Abbildung 4 ldap\_search\_2

Bei diesem ldapsearch wurde nach der User ID des User thomas.stedronsky gesucht.

ldapsearch -h 127.0.0.1 -p 389 -D "cn=thomas.stedronsky,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com" "cn=group.\*" memberUid

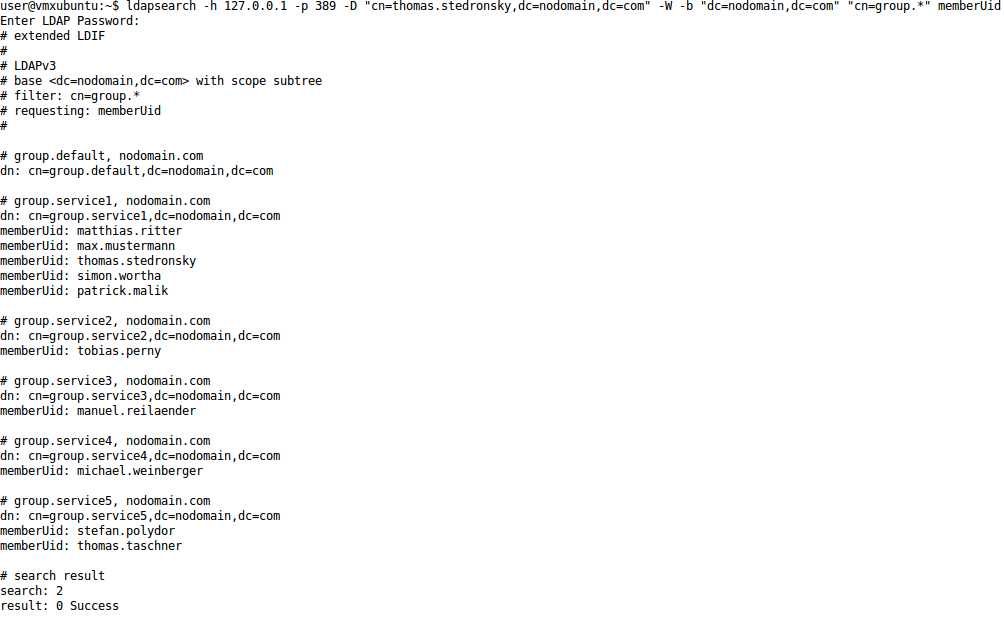


Abbildung 5 ldap\_search\_3

Bei diesem ldapsearch wird nach allen Gruppen gesucht, dies geschieht mit group.\*. Die User werden dann wieder anhand der MemberUid ausgegeben.

## LDAP Modify

ldapmodify -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W

dn: cn=group.service1,dc=nodomain,dc=com

changetype: modify

replace: description

description: testtest

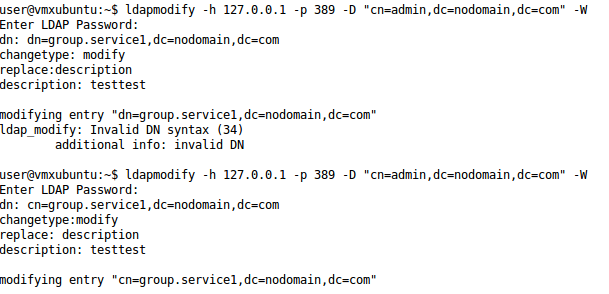


Abbildung 6 ldap\_modify\_1

Bei diesem Modify wurde die Description von der Gruppe Service1 verändert.

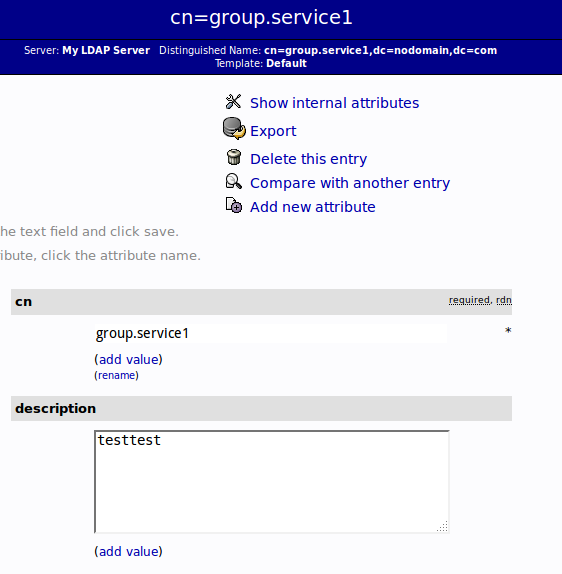


Abbildung 7 ldap\_modify\_result\_1

Hier sieht man die veränderte Description im LDAP-Webinterface

ldapmodify -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W

dn: cn=tobias.perny,dc=nodomain,dc=com

changetype: modify

replace: sn

description: Testtest

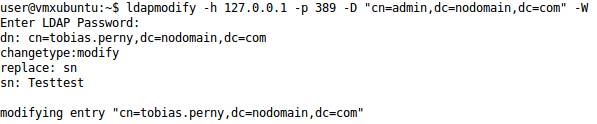


Abbildung 8 ldap\_modify\_1



Abbildung 9 ldap\_modify\_result\_2

Der Nachname(„Second Name“) von dem User tobias.perny wurde geändert.

## LDAP Änderung ohne Admin Rechte

## Brute-Force

In einer Schulübung wurde eine Brute-Force Attacke auf einen Account des LDAP Systems durchgeführt. . Hierbei wurde ein Algorithmus entwickelt der das Passwort eines Accounts knacken soll. Es wurden 4-, 8-, 16, und 64-stellige Passwörter vergeben. Es war uns leider aus zeitlichen Gründen und des TGM-Netzes nicht möglich ein Passwort herauszufinden.

# Quellen

[1] JNDI Tutorial, Oracle, docs.oracle.com/javase/tutorial/displayCode.html?code=http://docs.oracle.com/javase/tutorial/jndi/ldap/examples/Simple.java, zuletzt besucht am 08.01.2016

# Abbildungsverzeichnis

[Abbildung 1 create\_group 4](#_Toc440014122)

[Abbildung 2 create\_user 4](#_Toc440014123)

[Abbildung 3 ldap\_search\_1 5](#_Toc440014124)

[Abbildung 4 ldap\_search\_2 5](#_Toc440014125)

[Abbildung 5 ldap\_search\_3 6](#_Toc440014126)

[Abbildung 6 ldap\_modify\_1 7](#_Toc440014127)

[Abbildung 7 ldap\_modify\_result\_1 7](#_Toc440014128)

[Abbildung 8 ldap\_modify\_1 8](#_Toc440014129)

[Abbildung 9 ldap\_modify\_result\_2 8](#_Toc440014130)