

---

# Laborprotokoll

## Authentifizierung & Autorisierung

---

Systemtechnik Labor  
5BHITT 2015/16, Gruppe C

Thomas Taschner

Note:  
Betreuer: Thomas Micheler

Version 1.0  
Begonnen am 8. Januar 2016  
Beendet am 22. Januar 2016

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Ziele . . . . .	1
1.2	Voraussetzungen . . . . .	1
1.3	Aufgabenstellung . . . . .	1
<b>2</b>	<b>Ergebnisse</b>	<b>3</b>
2.1	Einrichten der VM . . . . .	3
2.1.1	Installation von LDAP . . . . .	3
2.1.2	Installation von phpLDAPAdmin . . . . .	3
2.1.3	Konfiguration von phpLDAPAdmin . . . . .	3
2.1.4	Konfiguration von Apache . . . . .	3
2.2	Anlegen von 5 Gruppen und 10 Personen . . . . .	4
2.2.1	Erstellen einer Gruppe . . . . .	5
2.2.2	Erstellen einer Person . . . . .	6
2.2.3	Gruppenzuweisungen . . . . .	7
2.3	LDAPSEARCH . . . . .	7
2.3.1	Erklärung des Befehls . . . . .	7
2.3.2	LDAPSEARCH 1 . . . . .	8
2.3.3	LDAPSEARCH 2 . . . . .	8
2.3.4	LDAPSEARCH 3 . . . . .	9
2.4	LDAPMODIFY . . . . .	9
2.4.1	LDAPMODIFY 1 . . . . .	9
2.4.2	LDAPMODIFY 2 . . . . .	10
2.5	Authentifizierung . . . . .	11
2.6	Autorisierung . . . . .	11
2.7	Änderungen mit bestimmtem User . . . . .	12

# 1 Einführung

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

## 1.1 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 durchgeführt.

## 1.2 Voraussetzungen

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

## 1.3 Aufgabenstellung

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

Bewertung: 16 Punkte

- Dokumentation der einzelnen Arbeitsschritte im Protokoll (2 Punkte)  
Anlegen von 5 Gruppen und 10 User Accounts (6 Punkte) (wenn fremdes LDAP-Service verwendet wird, dann Dokumentation von 3 LDAPSEARCH und 2 LDAPMODIFY Befehlen)
- Authentifizierung (4 Punkte)
- Autorisierung (4 Punkte)
- Wie ist eine LDAP Änderung möglich mit bestimmten Benutzer (ungleich admin)?
- Brute Force Implementierung

## 2 Ergebnisse

### 2.1 Einrichten der VM

#### 2.1.1 Installation von LDAP

```
1 sudo apt-get update
  sudo apt-get install slapd ldap-utils
```

Listing 1: Installation von LDAP

Nun erfolgt die Konfiguration des Pakets slapd. Hierzu muss folgendes bei der Installation eingestellt werden:

```
3 DNS domain name: nodomain.com
  Organization name: nodomain
  Administrator password: user
  Database backend: hdb
```

Listing 2: Konfiguration von slapd

#### 2.1.2 Installation von phpLDAPAdmin

Die Installation von phpLDAPAdmin lässt sich mit dem folgenden Befehl bewerkstelligen:

```
1 sudo apt-get install phpldapadmin
```

Listing 3: Installation von phpLDAPAdmin

#### 2.1.3 Konfiguration von phpLDAPAdmin

In diesem Schritt wird die Datei `/etc/phpldapadmin/config.php` um folgende Zeilen ergänzt:

```
4 $servers->setValue('server','host','localhost');
  $servers->setValue('server','base',array('dc=nodomain,dc=com'));
  $servers->setValue('login','bind_id','cn=admin,dc=nodomain,dc=com');
  $config->custom->appearance['hide_template_warning'] = true;
```

Listing 4: Konfiguration von phpLDAPAdmin

Eine Konfiguration für eine SSL gesicherte Verbindung wird hier nicht vorgenommen.

#### 2.1.4 Konfiguration von Apache

Abschließend muss nur noch ein Alias-Eintrag in der Datei `/etc/apache2/mods-enabled/alias.conf` erfolgen.

```
1 <IfModule alias_module>
  ...
  Alias /ldap /usr/share/phpldapadmin/htdocs
  ...
</IfModule>
```

Listing 5: Hinzufügen eines Alias-Eintrags

Wichtig dabei ist, dass sich der Eintrag innerhalb des IfModule Tags befindet. PhpLDAPAdmin sollte nun unter `http://localhost/ldap` erreichbar sein.

## 2.2 Anlegen von 5 Gruppen und 10 Personen

Zunächst muss ein erfolgreicher Login als admin erfolgen, die Startseite sollte angezeigt werden.



Abbildung 1: phpLDAPadmin Startseite

Im Frame My LDAP Server muss auf der linken Seite nun der Verzeichnisbaum aufgeklappt werden.

### 2.2.1 Erstellen einer Gruppe

Hierzu erfolgt ein Klick auf den Menüeintrag “Neuen Eintrag erzeugen“. Im rechten Frame muss nun eine Vorlage für das neu zu erstellende Objekt ausgewählt werden. Für eine Gruppe wird die Vorlage “Allgemein: POSIX-Gruppe“ ausgewählt.

Für das Erzeugen eines neuen Eintrages muss lediglich nur der Gruppenname eingetragen werden. Es wurde mit dem Betreuer eine bestimmte Namensgebung der Gruppen ausgemacht. Jede Gruppe muss den Namen `group.service[n]` tragen, wobei `n` eine Zahl zwischen 1 und 5 ist. Die Gruppen-ID wird selbstständig vergeben. Optional können auch bereits vorhandene Benutzer der Gruppe hinzugefügt werden. Abschließend kann der Eintrag per Knopfdruck abgespeichert werden.

The screenshot shows a web interface titled "Erzeuge einen neuen Eintrag" (Create a new entry). Below the title, it displays "Server: My LDAP Server" and "Container: dc=nodomain,dc=com". The selected template is "Vorlage: Generic: Posix Group (posixGroup)". The main heading is "New Posix Group (Step 1 of 1)".

The form has three main sections:

- GID-Nummer**: A text input field containing "505". A link "alias, notwendig, Hinweis, ro" is visible.
- Gruppe**: A text input field containing "group.service1". A link "alias, notwendig, rdn" is visible.
- Users**: A list of users with checkboxes next to their names and LDAP IDs. The users are:
  - ☐ thomas.taschner (ttas)
  - ☐ stefan.polydor (spol)
  - ☐ michael.weinberger (mwei)
  - ☐ thomas.micheler (tmic)
  - ☐ sebastian.steinkellner (sste)
  - ☐ manuel.reilaender (mrei)
  - ☐ tobi.perny (tper)
  - ☐ patrick.malik (pmal)
  - ☐ simon.wortha (swor)
  - ☐ mathias.ritter (mrit)A link "alias, Hinweis" is visible.

At the bottom, there is a button labeled "Erzeuge einen neuen Eintrag".

Abbildung 2: Erstellen einer POSIX-Gruppe

**Anmerkung:** Im Laufe der Übungsstunde bekamen wir von unserem Betreuer den Tipp die Gruppen zuerst anzulegen, da die Benutzer nachher beim Erstellen einer Gruppe zugewiesen werden müssen.

## 2.2.2 Erstellen einer Person

Auch hier erfolgt ein Klick auf den Menüeintrag “Neuen Eintrag erzeugen“. Im rechten Frame muss nun eine Vorlage für das neu zu erstellende Objekt ausgewählt werden. Für ein Benutzerkonto wird die Vorlage “Allgemein: Benutzerkonto“ ausgewählt.

Für das Erzeugen eines neuen Eintrages müssen der Benutzername (hier als Üblicher Name dargestellt), die Gruppenzuweisung, das Heimverzeichnis, der Nachname und die Benutzer-ID eingetragen werden. Auch hier wurde mit dem Betreuer ausgemacht, dass der Benutzername des Benutzers im Format vorname.nachname eingetragen wird. Optional können noch ein Vorname, eine Login shell und ein Passwort festgelegt werden. Die Benutzer-ID in numerischer Form wird automatisch generiert. Abschließend kann der Eintrag per Knopfdruck abgespeichert werden.

Erzeuge einen neuen Eintrag

Server: My LDAP Server Container: dc=nodomain,dc=com  
Vorlage: Generic: User Account (posixAccount)

New User Account (Step 1 of 1)

Üblicher Name alias, notwendig, rdn  
max.mustermann \*

Vorname alias  
Max

GID-Nummer alias, notwendig, Hinweis  
group.service1 \*

Heimverzeichnis alias, notwendig  
/home/users/mmustermann \*

Last name alias, notwendig  
Mustermann \*

Login shell alias  
\*

Passwort alias, Hinweis  
md5 (bestätigen)  
Prüfe Kennwort...

UID Number alias, notwendig, Hinweis, ro  
1010

User ID alias, notwendig  
mmustermann \*

Erzeuge einen neuen Eintrag

Abbildung 3: Erstellen eines Benutzerkontos

**Anmerkung:** Der Benutzername kann automatisch anhand des Vor- und Nachnamen generiert werden. Es wäre von Vorteil beide Werte vorher einzutragen.



### 2.2.3 Gruppenzuweisungen

Um der Aufgabenstellung gerecht zu werden, wurden 5 Gruppen und 10 Benutzer angelegt. Welcher Benutzer welcher Gruppe zugeteilt wurde, kann der folgenden Tabelle entnommen werden.

	service. group1	service. group2	service. group3	service. group4	service. group5
patrick.malik				X	
thomas.micheler		X			
tobias.perny				X	
stefan.polydor	X				
manuel.reilaender			X		
matthias.ritter					X
sebastian.steinkellner			X		
thomas.taschner	X				
michael.weinberger		X			
simon.wortha					X

Tabelle 1: Gruppenzuweisungen der Benutzer

Der Benutzer muss der Gruppe hinzugefügt werden (memberUid setzen)!

## 2.3 LDAPSEARCH

### 2.3.1 Erklärung des Befehls

```
ldapsearch -h 192.168.0.8 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com"
```

Listing 6: LDAPSEARCH Befehl

Zur Suche sind die folgenden Parameter relevant:

- h... Adresse des Servers, auf dem der LDAP-Dienst ausgeführt wird
- p... Port, auf dem der LDAP-Dienst erreichbar ist
- D... Anmeldedaten für den LDAP-Dienst (distinguished name)
- W... Passwort
- b... Startpunkt für die Suche [1]

### 2.3.2 LDAPSEARCH 1

```
ldapsearch -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
Enter LDAP Password:
# extended LDIF
4 #
# LDAPv3
# base <> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
9 #
# search result
search: 2
result: 32 No such object
14 # numResponses: 1
```

Listing 7: LDAPSEARCH 1

Es wurde lediglich nur ein Bind auf unsere lokale Instanz durchgeführt.

### 2.3.3 LDAPSEARCH 2

```
ldapsearch -h 192.168.188.34 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
Enter LDAP Password:
# extended LDIF
5 #
# LDAPv3
# base <> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
10 #
# search result
search: 2
result: 32 No such object
15 # numResponses: 1
```

Listing 8: LDAPSEARCH 2

Es wurde lediglich nur ein Bind auf unsere entfernte Instanz durchgeführt.

### 2.3.4 LDAPSEARCH 3

```
ldapsearch -h 192.168.188.34 -p 389 -D "cn=thomas.taschner,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com"
"
Enter LDAP Password:
# extended LDIF
#
5 # LDAPv3
# base <dc=nodomain,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
10 # nodomain.com
dn: dc=nodomain,dc=com
objectClass: top
objectClass: dcObject
15 objectClass: organization
o: nodomain
dc: nodomain

# admin, nodomain.com
20 dn: cn=admin,dc=nodomain,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
25
# group.service1, nodomain.com
dn: cn=group.service1,dc=nodomain,dc=com
gidNumber: 500
cn: group.service1
30 objectClass: posixGroup
objectClass: top

...

35 # search result
search: 2
result: 0 Success

# numResponses: 18
40 # numEntries: 17
```

Listing 9: LDAPSEARCH 3

Eine externe Suche, die uns sämtliche Einträge der Domäne ausgibt.

## 2.4 LDAPMODIFY

### 2.4.1 LDAPMODIFY 1

```
ldapmodify -h 192.168.188.34 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
Enter LDAP Password:
dn: cn=group.service1,dc=nodomain,dc=com
4 changetype: modify
replace: description
description: test

modifying entry "cn=group.service1,dc=nodomain,dc=com"
```

Listing 10: LDAPMODIFY 1

Ändert die Beschreibung der Gruppe group.service1 auf test. Nun wird überprüft, ob die Änderung übernommen wurde.

```
1 ldapsearch -h 192.168.188.34 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W -b "cn=group.service1,dc=
  nodomain,dc=com"
  ...
  # group.service1, nodomain.com
  dn: cn=group.service1,dc=nodomain,dc=com
  gidNumber: 500
6  cn: group.service1
  objectClass: posixGroup
  objectClass: top
  description: test
  ...
```

Listing 11: LDAPMODIFY 1 Check

## 2.4.2 LDAPMODIFY 2

```
ldapmodify -h 192.168.188.34 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
Enter LDAP Password:
dn: cn=thomas.taschner,dc=nodomain,dc=com
4 changetype: modify
  add: telephoneNumber
  telephoneNumber: 0123456789

modifying entry "cn=thomas.taschner,dc=nodomain,dc=com"
```

Listing 12: LDAPMODIFY 2

Fügt dem Benutzer thomas.taschner das Attribut telephoneNumber hinzu. Nun wird überprüft, ob die Änderung übernommen wurde.

```
1 ldapsearch -h 192.168.188.34 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W -b "cn=thomas.taschner,dc=
  nodomain,dc=com"
  ...
  # thomas.taschner, nodomain.com
  dn: cn=thomas.taschner,dc=nodomain,dc=com
  givenName: Thomas
6  gidNumber: 500
  homeDirectory: /home/users/ttaschner
  sn: Taschner
  loginShell: /bin/sh
  objectClass: inetOrgPerson
11 objectClass: posixAccount
  objectClass: top
  uidNumber: 1000
  uid: ttaschner
  cn: thomas.taschner
16 userPassword:: e0IENX1mcTdNOVIEYUVHd01RL01Ja1VGcFZnPT0=
  telephoneNumber: 0123456789
  ...
```

Listing 13: LDAPMODIFY 2 Check

## 2.5 Authentifizierung

Hierzu wurde der Code [2] übernommen und entsprechend an unsere Bedürfnisse angepasst. Sollte die Autorisierung klappen, so wird ein OK zurückgegeben. Sollte sie fehlschlagen, so wird ein NOK zurückgegeben. Zu Testzwecken werden beide Szenarien ausprobiert.

```
1 package ldap;

import java.util.Hashtable;

import javax.naming.*;
6 import javax.naming.directory.*;

public class LDAPAuthentication {

    private DirContext ctx;
11 private String username;
    private final String ldapSearchBase = "dc=nodomain,dc=com";

    public LDAPAuthentication(String host, String username, String pw, int port) {
        this.username = "cn=" + username + ",dc=nodomain,dc=com";
16 Hashtable<String, String> env = new Hashtable<String, String>();
        env.put(Context.INITIAL_CONTEXT_FACTORY,
                "com.sun.jndi.ldap.LdapCtxFactory");
        env.put(Context.PROVIDER_URL, "ldap://" + host + ":" + port);

21 env.put(Context.SECURITY_AUTHENTICATION, "simple");
        env.put(Context.SECURITY_PRINCIPAL, this.username);
        env.put(Context.SECURITY_CREDENTIALS, pw);

        try {
26 ctx = new InitialDirContext(env);
            System.out.println("Authentifizierung: OK");
        } catch (NamingException ne) {
            System.out.println("Authentifizierung: NOK");
            System.exit(1);
        } catch (Exception e) {
31 System.exit(1);
        }
    }

36 public static void main(String[] args) {
    new LDAPAuthentication("192.168.188.34", "thomas.taschner", "thomas.taschner", 389);
    new LDAPAuthentication("192.168.188.34", "thomas.taschner", "passwort123", 389);
}

41 }
```

Listing 14: Java Code zur LDAP Authentifizierung

## 2.6 Autorisierung

Hier wurde der Code angepasst und erweitert. Übernommen wurde er von hier [2] [3]. Es wird überprüft, ob der Benutzer Mitglied einer bestimmten Gruppe und daher berechtigt ist den Service zu nutzen. SearchControls ermöglicht es einem, in einem LDAP-Verzeichnis zu suchen. Sollte die Gruppe das Attribut memberuid enthalten, so wird OK ausgegeben. Sollte dies nicht der Fall sein, so wird NOK ausgegeben.

```
package ldap;

import javax.naming.Context;
4 import javax.naming.NamingEnumeration;
import javax.naming.NamingException;
import javax.naming.directory.*;
```

```

import java.util.Hashtable;

9
public class LDAPAuthorization {

    private DirContext ctx;
    private String username;
14    private String ldapSearchBase = "dc=nodomain,dc=com";

    public LDAPAuthorization(String host, String username, String password, int port, String group) {
        this.username = "cn=" + username + ",dc=nodomain,dc=com";
        Hashtable env = new Hashtable();
19        env.put(Context.INITIAL_CONTEXT_FACTORY,
                "com.sun.jndi.ldap.LdapCtxFactory");
        env.put(Context.PROVIDER_URL, "ldap://" + host + ":" + port);

        env.put(Context.SECURITY_AUTHENTICATION, "simple");
24        env.put(Context.SECURITY_PRINCIPAL, this.username);
        env.put(Context.SECURITY_CREDENTIALS, password);

        try {
            ctx = new InitialDirContext(env);
            System.out.println("Authentifizierung: OK");
        } catch (NamingException ne) {
            System.out.println("Authentifizierung: NOK");
            System.exit(1);
        }

34        try {
            SearchControls searchControls = new SearchControls();
            searchControls.setSearchScope(SearchControls.SUBTREE_SCOPE);
            searchControls.setTimeLimit(30000);

39            // Ueberpruefung ob user in der angegeben Gruppe ist
            NamingEnumeration<?> namingEnum = ctx.search("cn=" + group + ",dc=nodomain,dc=com", "(
                objectclass=posixGroup)", searchControls);

            while (namingEnum.hasMore ()) {
44                SearchResult result = (SearchResult) namingEnum.next ();
                Attributes attrs = result.getAttributes();
                System.out.println("Authorization: " + group + " " + (attrs.get("memberUID") != null &&
                    attrs.get("memberUid").contains(username) ? "OK" : "NOK"));
            }

49        } catch (NamingException ne) {
            System.out.println("NOK - Authorization");
        }
    }

54    public static void main(String[] args) throws NamingException {
        new LDAPAuthorization("10.0.104.73", "thomas.taschner", "thomas.taschner", 389, "group.service1"
        );
    }
}

```

Listing 15: Java Code zur LDAP Autorisierung

## 2.7 Änderungen mit bestimmtem User

Dies lässt sich mit Hilfe einer Access Control List umsetzen. Es lassen sich bestimmte Rechte für Benutzer und Gruppen zuteilen.

## Literatur

- [1] B.2. using ldapsearch. [https://www.centos.org/docs/5/html/CDS/ag/8.0/Finding\\_Directory\\_Entries-Using\\_ldapsearch.html](https://www.centos.org/docs/5/html/CDS/ag/8.0/Finding_Directory_Entries-Using_ldapsearch.html). Zuletzt besucht: 14.01.2016.
- [2] How do a ldap search/authenticate. <http://stackoverflow.com/questions/2172831/how-do-a-ldap-search-authenticate-against-this-ldap-in-java>. Zuletzt besucht: 14.01.2016.
- [3] Ldap command-line tools. [https://docs.oracle.com/cd/B10501\\_01/network.920/a96579/comtools.htm#63](https://docs.oracle.com/cd/B10501_01/network.920/a96579/comtools.htm#63). Zuletzt besucht: 14.01.2016.

## Tabellenverzeichnis

1	Gruppenzuweisungen der Benutzer . . . . .	7
---	---	---

## Listings

1	Installation von LDAP . . . . .	3
2	Konfiguration von slapd . . . . .	3
3	Installation von phpLDAPAdmin . . . . .	3
4	Konfiguration von phpLDAPAdmin . . . . .	3
5	Hinzufügen eines Alias-Eintrags . . . . .	3
6	LDAPSEARCH Befehl . . . . .	7
7	LDAPSEARCH 1 . . . . .	8
8	LDAPSEARCH 2 . . . . .	8
9	LDAPSEARCH 3 . . . . .	9
10	LDAPMODIFY 1 . . . . .	9
11	LDAPMODIFY 1 Check . . . . .	10
12	LDAPMODIFY 2 . . . . .	10
13	LDAPMODIFY 2 Check . . . . .	10
14	Java Code zur LDAP Authentifizierung . . . . .	11
15	Java Code zur LDAP Autorisierung . . . . .	11

## Abbildungsverzeichnis

1	phpLDAPAdmin Startseite . . . . .	4
---	-----------------------------------	---

2	Erstellen einer POSIX-Gruppe . . . . .	5
3	Erstellen eines Benutzerkontos . . . . .	6