JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

DezSys04

Systemtechnik-Labor
5BHIT 2015/16, Gruppe Z

Michael Weinberger

Version 1.0
Begonnen am 08. Januar 2016
Beendet am 14. Januar 2016

Note:
Betreuer: Th.Micheler

Inhaltsverzeichnis

1 Einführung

Diese Übung soll zur Vertiefung der Begriffe Äuthentifizierung 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 fuer 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 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)

2 Dokumentation der Arbeitsschritte

2.1 Grundkonfiguration

Folgendes Textfile von Prof. Micheler beschreibt das Aufsetzen eines OpenLDAP-Servers, die Grundkonfiguration, Grundlagen in LDAPSEARCH sowie LDAPMODIFY und listet einige weiterführende Links auf.

```
Installation LDAP:
------------------
sudo apt-get update
sudo apt-get install slapd ldap-utils

sudo dpkg-reconfigure slapd
> DNS domain name: nodomain.com
> Organization name: nodomain
> Administrator password: user
> Database backend: hdb

Installation phpLDAPadmin:
--------------------------
sudo apt-get install phpldapadmin

Configuration phpLDAPadmin:
---------------------------
sudo gedit /etc/phpldapadmin/config.php

$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;

SSL configuration not performed!

Configuration Apache:
---------------------
/etc/apache2/mods-enabled/alias.conf: following line added
Alias /ldap /usr/share/phpldapadmin/htdocs

Link to phpLDAPadmin:
---------------------
http://localhost/ldap

Modify LDAP Directory:
----------------------
Add new Posix Group: group.default
Add new Posix Group: group.service1
Add new Generic User Account: max.mustermann
Add max.mustermann to group.service1

LDAPSEARCH Commandline Tool / Local:
------------------------------------
ldapsearch -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
ldapsearch -h 127.0.0.1 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W

LDAPSEARCH Commandline Tool / Remote:
-------------------------------------
ldapsearch -h 192.168.0.8 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
ldapsearch -h 192.168.0.8 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com"
ldapsearch -h 192.168.0.8 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W -b "cn=group.service2,dc=
    nodomain,dc=com" memberUid
ldapsearch -h 192.168.0.8 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com" "
    cn=group.*" memberUid
ldapsearch -h 192.168.0.8 -p 389 -D "cn=max.mustermann,dc=nodomain,dc=com" -W -b "dc=nodomain,dc=com" "(
    objectclass=PosixGroup)"
```

Listing 1: Grundkonfiguration


```
LDAPMODIFY Commandline Tool / Remote:
_____________________________________

ldapmodify -h 192.168.0.8 -p 389 -D "cn=admin,dc=nodomain,dc=com" -W
dn: cn=group.service1,dc=nodomain,dc=com
changetype: modify
replace: description
description: test


Links:
______

http://docs.oracle.com/javase/tutorial/jndi/index.html
http://www.stefan-seelmann.de/media/presentations/JUGM2008_JavaUndLDAP.pdf
https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-openldap-and-phpldapadmin-
    on-an-ubuntu-14-04-server
```

Listing 2: Grundkonfiguration

2.2 Anlegen von 5 Gruppen und 10 User-Accounts

In der beschriebenen Testumgebung ist die phpLDAPadmin-Oberfläche via *http://localhost/ldap/* aufzurufen. Im Login-Screen meldet man sich per Credentials *cn=admin,dc=nodomain,dc=com* und *user* an. In der Adminoberfläche findet sich im linken Menü der Eintrag 'Create new entry here'. Eine Liste an Templates für den Erstellungsprozess wird angezeigt, wir wählen zuerst 'Generic: Posix Group' aus. Die GID-Nummer wird automatisch generiert, der Gruppe kann auch ein Name gegeben werden, in unserem Fall 'service1' bis 'service5'.

Um einen User zu erstellen findet sich unter 'Create new entry here' der Eintrag 'Generic: User Account'. Relevant bei der Eingabe ist der Vor- und Nachname, die GID-Nummer (zugehörige Gruppe) und das Passwort, der Rest wird automatisch generiert aus den bereitgestellten Daten, kann gegebenenfalls trotzdem noch angepasst werden.

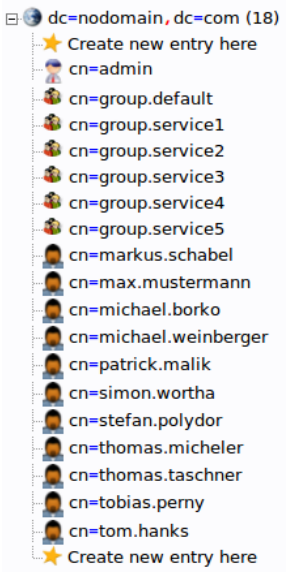

Abbildung 1: 10 User-Accounts, 5 Posix-Groups

2.3 LDAPSEARCH und LDAPMODIFY

2.3.1 LDAPSEARCH 1

```
1 ldapsearch -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -w user
```

Listing 3: 1.Befehl

Mit diesem Befehl wird erfolgreich ein Bind auf den Localhost-LDAP-Server (Port 389/TLS) durchgeführt, ohne Suchanfrage und ohne nennenswerte Ausgabe.

2.3.2 LDAPSEARCH 2

```
ldapsearch -h 192.168.128.136 -p 389 -D "cn=admin,dc=nodomain,dc=com" -w user
```

Listing 4: 2.Befehl

Gleiches Vorgehen wie bei LDAPSEARCH 1, jedoch wird nun von 'außen' auf den LDAP-Server zugegriffen.

2.3.3 LDAPSEARCH 3

```
ldapsearch -h 192.168.128.136 -p 389 -D "cn=admin,dc=nodomain,dc=com" -w user -b "dc=nodomain,dc=com"
```

Listing 5: 3.Befehl

Wieder ein Zugriff von außen, mithilfe von -b kann der Startpunkt der Suche angegeben werden, hier wird der gesamte Inhalt unserer Domäne ausgegeben.

2.3.4 LDAPMODIFY 1

```
ldapmodify -h 192.168.0.8 -p 389 -D "cn=admin,dc=nodomain,dc=com" -w user
dn: cn=michael.weinberger,dc=nodomain,dc=com
changetype: modify
4 replace: sn
sn: Mueller
```

Listing 6: 4.Befehl

Zugriff von außen, der Nachname ('sn') wird per Befehl auf den Namen 'Müller' gesetzt.

2.3.5 LDAPMODIFY 2

```
ldapmodify -h 127.0.0.1 -p 389 -D "cn=admin,dc=nodomain,dc=com" -w user
-f /home/user/Documents/SYT/data.ldif
```

Listing 7: 5.Befehl

Lokaler Aufruf, liest anstatt von stdin die Änderungsinformationen aus einem LDIF-File (LDAP Data Interchange Format), das derselben Struktur entsprechen muss.

3 Authentifizierung

Im Oracle JNDI Tutorial ist der Beispielcode implementiert worden, der für unsere Anwendung relevant ist. [1]
Via JOptionPane wird der User aufgefordert die Parameter einzugeben, ist daher nicht statisch und für jeden LDAP-Server, der unserem Aufbau ähnelt einsetzbar. Sofern der Bind auf den LDAP-Server geglückt ist, wird ein 'OK' ausgegeben, andernfalls 'NOK'.

4 Autorisierung

Um auch zu überprüfen, dass der User autorisiert ist (Mitglied einer Gruppe) wurde Sourcecode eines Forums bezogen [2]. Über 'SearchControls' (javax.naming.directory.SearchControls) kann in einem LDAP-Verzeichnis gesucht werden. Enthält die Gruppe das Attribut 'memberuid', so wird OK ausgegeben, andernfalls NOK.

5 LDAP-Änderung mit bestimmten User

Änderungen ohne Adminrechte lassen sich per ACL (Access Control List) definieren. In jener kann man bestimmte Rechte für Benutzer, Gruppen zuteilen. Diese Konfiguration muss in der slapd-Konfigurationsdatei */etc/ldap/slap.d* vorgenommen werden.

```
access to dn.subtree="dc=example,dc=com" attrs=homePhone
        by self write
        by dn.children="dc=example,dc=com" search
        by peername.regex=IP:10\..+ read
    access to dn.subtree="dc=example,dc=com"
        by self write
        by dn.children="dc=example,dc=com" search
        by anonymous auth
```

Listing 8: ACL

6 Quellen

[1]: http://docs.oracle.com/javase/tutorial/jndi/ldap/examples/Simple.java, Oracle Documentation, zuletzt abgerufen am 14.01.2016
[2]: http://stackoverflow.com/questions/2172831/how-do-a-ldap-search-authenticate-against-this-ldap-in-java, leider Gottes Stack Overflow, zuletzt abgerufen am 14.01.2016

Listings

Abbildungsverzeichnis