JUST DO IT
HÖHERE ABTEILUNG FÜR INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Laborprotokoll

DezSys04: Authentifizierung & Autorisierung

Systemtechnik Labor
5BHITT 2015/16, Gruppe X

Stefan Erceg

Version 1.0

Note:

Betreuer: Prof. Micheler

Begonnen am 20. November 2015

Beendet am 26. November 2015

Inhaltsverzeichnis

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 eine fertig konfiguriertes 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 Anlegen von 5 Gruppen und 10 Personen

Um die Gruppen und Personen anzulegen, wurde im Browser die PHP-LDAP-Admin Oberfläche durch die URL http://10.0.106.159/phpldapadmin aufgerufen. Nachdem man sich als admin (Passwort: user) angemeldet hat, kann man die Gruppen bzw. Personen erstellen.

Möchte man Gruppen erstellen, muss man sich folgendermaßen durchklicken:

- „Neuen Eintrag erzeugen"
- Vorlagen: Allgemein: POSIX-Gruppe
- Gruppe = group.service[groupnr]
  - groupnr = 1, 2, 3, 4, 5
- Button „Erzeuge einen neuen Eintrag" klicken
- Button „Anwenden" klicken

Um Personen zu erstellen und diese den Gruppen zuzuweisen (jeder Gruppe werden 2 Personen hinzugefügt) muss man folgendes durchführen:

- „Neuen Eintrag erzeugen"
- Vorlagen: Allgemein: Benutzerkonto
- Common Name: vorname.nachname
  - z.B.: stefan.erceg
- Vorname eingeben
- GID-Nummer: jeweilige Gruppennummer, zu der man die Person hinzufügen möchte
  - z.B.: group.service1
- Nachname eingeben
- Passwort: für jede Person „12345"
- Button „Erzeuge einen neuen Eintrag" klicken
- Button „Anwenden" klicken

Damit die Personen dann schlussendlich tatsächlich einer bestimmten Gruppe zugewiesen sind, muss folgendes durchgeführt werden:

- Klick auf jeweilige Gruppe
  - z.B.: cn=group.service1
- „Neues Attribut hinzufügen"
- Attribut hinzufügen: memberUid
- User Name angeben
- „Update Object" klicken
- Änderungen übernehmen: erneut „Update Object" klicken
- Bei memberUid können dann noch unter „modify group members" weitere Personen zu der jeweiligen Gruppe hinzugefügt werden.

2.2 Implementierung der Authentifizierung

Für die Authentifizierung wurde ein Example aus dem offiziellen Java Tutorial von Oracle genommen [1]. In dem Example wird ein Kontext erstellt und überprüft, ob sich ein bestimmter User aus einem Verzeichnisdienst authentifizieren kann.

Zum Aufbereiten des Kontexts wird eine Hash-Tabelle erstellt. In diese werden Daten, wie z.B. der Hostname, die Authentifizierungs-Methode und die Daten bezüglich des Users, hinzugefügt:

```
public static final String HOST = "ldap://10.0.106.159:389";

Hashtable<String, Object> env = new Hashtable<String, Object>(11);
env.put(Context.PROVIDER_URL, HOST);
```

Danach wird der Versuch zum Erstellen des Kontexts durchgeführt. Falls der Kontext erstellt werden konnte, wird `Authentifizierung: OK` in die Konsole geschrieben, falls es jedoch zu einer NamingException kommt wird `Authentifizierung: NOK` angezeigt. Der Kontext muss zum Schluss auch geschlossen werden:

```
try {
      DirContext ctx = new InitialDirContext(env);
      System.out.println("Authentifizierung: OK");
      ctx.close();

} catch (NamingException e) {
      System.out.println("Authentifizierung: NOK");
}
```

2.3 Implementierung der Autorisierung

Bei der Autorisierung habe ich überprüft, ob sich der User „serceg" in der Gruppe „group.service1" befindet.

Dazu wird zu Beginn definiert, wo genau im Verzeichnisdienst nach dem User gesucht wird und wie lange die Suche erfolgen soll bis ein Timeout erfolgt:

```
SearchControls searchControls = new SearchControls();
searchControls.setSearchScope(SearchControls.SUBTREE_SCOPE);
searchControls.setTimeLimit(30000);
```

Um den User in der Gruppe "group.service1" dann tatsächlich zu finden, werden in einer while-Schleife die Attribute von jedem Search-Result überprüft. Falls die memberUid eines Results dem Usernamen „serceg" entspricht, konnte die Autorisierung erfolgreich durchgeführt werden:

```
if(attrs.get("memberUid").contains(USER_NAME))
    System.out.println("Autorisierung: OK");
else
    System.out.println("Autorisierung: NOK");
```

2.4 LDAP-Änderung mit bestimmten Benutzer durchführen

Damit ein bestimmter User ebenfalls LDAP-Änderungen durchführen kann, können sogenannte „Access Control Lists" (ACLs) definiert werden. Dabei können einem User Rechte für bestimmte Einträge und Attribute zugewiesen werden. Das Argument „access" besitzt folgende generelle Form:

```
<access directive> ::= access to <what>
    [by <who> [<access>] [<control>] ]+
```

„who" definiert, wem die Rechte verteilt werden sollen und „what" für welche Einträge und/oder Attribute dies zutrifft.

Folgende Tabelle zeigt, wie man die Argumente für die Benutzer, denen die Rechte zugewiesen werden sollen, definiert:

Argument	betroffene Benutzer
*	alle (sowohl anonyme, als auch authentifizierte) User
anonymous	anonyme bzw. nicht-authentifizierte User
users	authentifizierte User
self	User, die dem Zieleintrag zugeordnet sind
dn[.<basic-style>]=<regex>	User, auf die die Regular-Expression zutrifft
dn.<scope-style>=<DN>	User im Rahmen des DNs

[2, 3]

2.5 Brute-Force-Implementierung

Für diesen Task hat sich unsere Klasse in den beiden DezSys-Stunden in 4 Gruppen aufgeteilt. Jede Gruppe erstellte ein 4-, 8-, 16- und 64-stelliges Passwort und diese mussten von den anderen Gruppen mittels eines Brute-Force-Algorithmus geknackt werden. Ich war in der Gruppe von Hagen Fock, der den LDAP-Server zur Verfügung stellte.

Schlussendlich konnte meine Gruppe die Passwörter der anderen Gruppen nicht knacken, da das TGM-Netz zu diesem Zeitpunkt sehr langsam und ausgelastet war und zu viele Passwort-Möglichkeiten berechnet wurden. Es wurde jedoch ein korrekter Algorithmus zum Knacken der Passwörter durchgeführt.

3 Zeitaufwand

In der Laboreinheit ist es mir gelungen, die Gruppen und Personen beim LDAP-Server anzulegen und die Authentifizierung & Autorisierung fertig zu implementieren. Die Brute-Force-Implementierung wurde in den beiden DezSys-Unterrichtsstunden durchgeführt.

Um herauszufinden, wie eine LDAP-Änderung mit einem bestimmten Benutzer, der kein Admin ist, möglich ist, habe ich noch zusätzliche 40 Minuten in meiner Freizeit investiert.

4 Quellenangaben

[1] Oracle (1995, 2008).
JNDI: Java Tutorials Code Sample – Simple.java [Online]. Available at: http://docs.oracle.com/javase/tutorial/jndi/ldap/examples/Simple.java [zuletzt abgerufen am 20.11.2015]

[2] OpenLDAP Foundation (2011). Chapter 8 – Access Control [Online]. Available at: http://www.openldap.org/doc/admin24/access-control.html [zuletzt abgerufen am 26.11.2015]

[3] Stanford University (March 2015). OpenLDAP ACL Examples [Online]. Available at:https://itservices.stanford.edu/service/directory/aclexamples [zuletzt abgerufen am 26.11.2015]