PSA-Praktikum Blatt 8

LDAP

Team09

LDAP - Abkürzungen

- CN = Common Name
- OU = Organizational Unit
- DC = Domain Component

LDAP - Installation

Grundlegende Installation des default ldap servers für Ubuntu: slapd

```
sudo apt install slapd ldap-utils
```

Grundlegende Konfiguration

```
sudo dpkg-reconfigure slapd
```

Festlegen des Directory Information Tree (DIT)

```
dc=team09,dc=psa,dc=in,dc=tum,dc=de
```

LDAP - Daten hinzufügen

Befehle

```
ldapadd -x -D "cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de" -f file.ldif -W
ldapmodify -x -D "cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de" -f file.ldif -W
```

ldif Format

```
dn: uid=ge49vaz,ou=users,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
cn: Simon
sn: Heinrich
uid: ge49vaz
uidNumber: 1092
gidNumber: 1090
homeDirectory: /home/ge49vaz
loginShell: /bin/bash
gecos: Simon Heinrich
userPassword: XXXXXXX
```

LDAP - TLS

Für TLS nutzen wir das certtools für linux

```
sudo apt install gnutls-bin ssl-cert
```

Erstellen einer Template Datei für die CA unter `/etc/ssl/ca.info/` mit folgendem Inhalt:

```
cn = PSA TUM
ca
cert_signing_key
expiration_days = 3650
```

Selbstsignierten CA Zertifikat erzeugen:

```
sudo certtool --generate-self-signed \
--load-privkey /etc/ssl/private/mycakey.pem \
--template /etc/ssl/ca.info \
--outfile /usr/local/share/ca-certificates/mycacert.crt
```

CA Zertifikat zur Liste an vertrauenswürdigen CAs hinzuzufügen:

```
update-ca-certificates
```

LDAP - TLS

Ein certinfo.ldif Datei erzeugen

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/mycacert.pem

- add: olcTLSCertificateFile
  olcTLSCertificateFile: /etc/ldap/ldap01_slapd_cert.pem
- add: olcTLSCertificateKeyFile
  olcTLSCertificateKeyFile: /etc/ldap/ldap01_slapd_key.pem
```

Mit `ldapmodify` den LDAP Server anpassen

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

Testen

```
root@vmpsateam09-05:~# openssl s_client -connect 192.168.9.9:389 -starttls ldap
```

Teil des Outputs

```
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits

---

SSL handshake has read 2962 bytes and written 394 bytes
Verification: OK

---

New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
```

Organizational Units

Organizational Units (OU) dienen der Strukturierung der LDAP Dateneinträgen

OU allgemein anlegen

Organizational Units allgemein anlegen

```
dn: ou=users,dc=team09,dc=psa,dc=in,dc=tum,dc=de  #distinguished_name eintrag
objectclass: top                                  #classes with attributes
objectclass: organizationalUnit                   #classes with attributes
ou: users                                         #concrete attribute with value

dn: ou=groups,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectclass: top
objectclass: organizationalUnit
ou: groups

dn: ou=computers,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectclass: top
objectclass: organizationalUnit
ou: computers

dn: ou=psaou,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectclass: top
objectclass: organizationalUnit
ou: psaou
```

OU users

OU für alle Nutzerkennungen der Mitglieder des Praktikums

```
dn: uid=ge49vaz,ou=users,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
cn: Simon
sn: Heinrich
uid: ge49vaz
uidNumber: 1092
gidNumber: 1090
homeDirectory: /home/ge49vaz
loginShell: /bin/bash
gecos: Simon Heinrich
userPassword: XXXXXXX
```

OU groups

OU für Teams aus dem Praktikum

```
dn: cn=team09,ou=groups,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectClass: top
objectClass: posixGroup
gidNumber: 1090
```

OU computers

Nutzerkennungen für alle unsere VMs

```
dn: cn=vm05,ou=computers,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectClass: top
objectClass: person
cn: vm05
sn: VM 05 - Test Server
userPassword: XXXXXXX
```

OU psaou

Nutzerkennung für die Einträge aus dem CSV File

```
dn: Matrikelnummer=1622888953,ou=psaou,dc=team09,dc=psa,dc=in,dc=tum,dc=de
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
objectClass: psaPerson
uid: 1622888953
gidNumber: 10001
uidNumber: 8021
cn: Clarissa
sn: Attenberger
homeDirectory: /home/Attenberger
```

OU psaou

Gruppe für alle Nutzerkennung die wir aus dem CSV File eingelesen. Hier wird auch unser selbsterzeugtes Schema psaPerson genutzt.

```
usercertificate;binary:<file:///root/workspace/csv2ldif/testdata/public/1622888953.der
Nachname: Attenberger
Vorname: Clarissa
Geschlecht: m
Geburtsdatum: 02.01.88
Geburtsort: Pegnitz
Nationalitaet: Deutschland
Strasse: Ilinden Street nr. 145
PLZ: 53604
Ort: Muenchen DE
Telefon: 0455/67742938
Matrikelnummer: 1622888953
```

LDAP Schema

1. Custom Schema: psaPerson
2. Attribute die den Einträgen im CSV File entsprechen

LDAP Schema

Erzeugen einer `new.schema` Datei an

```
objectidentifier psaSchema 1.3.6.1.4.1.A.B  # Unique ObjectIdentifier OID for the scheme ⟶ A and B arbitary numbers for u
objectidentifier psaAttrs psaSchema:X       # OID for all Attributes ⟶ OID from scheme + ".X"
objectidentifier psaOCs psaSchema:Y         # OID for all ObjectClass definitions ⟶ OID from scheme + ".Y"

attributetype ( psaAttrs:1                  # new attributetype with OID psaAttrs + ".1"
NAME 'Nachname'                             # new name for the attributetype
DESC 'PSA Nachname Identifier'              # new description for the attributetype
EQUALITY caseIgnoreMatch                    # behavior for rules with equal name ⟶ here: ignore
SUBSTR caseIgnoreSubstringsMatch            # behavior for rules with similar name(substring) ⟶ here: ignore
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32} )  # attribute Type: String{field with 32 characters}
```

LDAP Schema

Erzeugen einer `new.schema` Datei an

```
attributetype ( psaAttrs:2
NAME 'Vorname'
DESC 'PSA Vorname Identifier'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32} )

#

# More Attributes here

#

## LDAP Schema
objectClass ( psaOCs:1                          # new objectClass with OID psaOCs + ".1"
NAME 'psaPerson'                                # new name for the objectClass
DESC 'Describe a PSA Person'                    # new description for the objectClass
SUP ( top ) AUXILIARY                           # Superior objectClass (here:top) ; type of objectClass here(AUXILIARY
MUST ( Matrikelnummer $ Name )                  # attributes that have to be filled
MAY ( Vorname $ Geschlecht $ Geburtsdatum $
Geburtsort $ Nationalitaet $ Strasse $ PLZ $ Ort $ Telefon ) ) # attributes that can be filled
```

LDAP Schema

Erzeugen einer `tmp.conf` Datei:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/inetorgperson.schema
include $path to new.schema file$
```

Erzeugen einer Test Config Umgebung des LDAP Servers

```
slaptest -f /$path$/test.conf -F /$path$/schema/tmp
```

Config Datei in das Produktionsverzeichnis kopieren und Server neustarten

```
cp /$path$/tmp/cn=config/cn=schema/cn={4}new.ldif /etc/ldap/slapd.d/cn=config/cn=schema/
systemctl restart slapd.service
```

Einlesen der CSV Datei

1. CSV Datei einlesen
2. Ausschreiben im richtigen Format in eine `ldif` datei
3. X.509 Zertifikat hinzufügen

Struktur der CSV Datei

Untersuchen der gegebenen Attribute und Daten in der CSV-Datei mit folgenden Befehl:

```
# head -n10 testdata/benutzerdaten.csv
"Name","Vorname","Geschlecht","Geburtsdatum","Geburtsort","Nationalit�t","Stra�e","PLZ","Ort","Telefon","M...nr"
Rimmelspacher,Michael,w,10.04.88,Wasserburg,TH,Neufahrner Str. 7,82031,Muenchen,02283-67794984,1574819974
Seidewitz,Paulo,w,23.02.84,Berlin,DE,Hauptstr. 13 d,81669,Muenchen,03008-89218323,1410829795
Hegenbartova,Charlotte,m,29.06.85,Muenchen,D,Kirchstr.4,82110,Sauerlach DE,04167/48999010,1533471176
Brueckner,Sara,m,14.08.84,Muenchen,DE,Semmelweisstr. 7,80805,Muenchen,0792/72430802,1632191735
Schrammel,Anatol,m,04.05.90,Muenchen,DE,Platanenweg 26,85551,Muenchen,06315/42473821,1948182970
Traykov,Jan,m,28.07.83,Frankfurt/Main,DE,Stiftsbogen 33,83123,Muenchen DE,0264-52279023,1694982524
Wang,Nora,m,02.11.84,Koesching,DE,Hohenwaldeckstr. 37,81379,Krumbach DE,07661/47518212,1194390678
Georgiev,Lukas,m,3.6.79,Dachau,deutsch,Helene-Mayer-Ring 7,80797,Muenchen,09015/84294955,1742634365
Shulman,Ferdinand,m,03.08.91,Heilbronn,DE,Obertal 27,38527,Muenchen,06119/38253096,1447636373
```

LdifEntry Klasse

Konstruktor und attribute

```
class LdifEntry:
    uidNum = ''
    attributes = {}
    userCertificatePath = ''

    def __init__(self, uidNum, attrNames, row):
        self.uidNum = uidNum

        # Replace Name attribute name with Nachname
        attrNames = ['Nachname' if item == 'Name' else item for item in attrNames]
        self.attributes = dict(zip(attrNames, row))

        self.userCertificatePath = CERTIFICATES + self.attributes["Matrikelnummer"] + ".der"
```

LdifEntry Klasse

```
    def __str__(self):
        entry = textwrap.dedent("""\
            dn: Matrikelnummer=%s,ou=psaou,dc=team09,dc=psa,dc=in,dc=tum,dc=de
            objectClass: posixAccount
            objectClass: shadowAccount
            objectClass: inetOrgPerson
            objectClass: psaPerson
            uid: %s
            gidNumber: 10001
            uidNumber: %s
            cn: %s
            sn: %s
            homeDirectory: /home/%s
            usercertificate;binary:<file://%s
        """%( self.attributes["Matrikelnummer"],
              self.attributes["Matrikelnummer"],
              ...
              self.userCertificatePath))

        for attrName, value in self.attributes.items():
            entry = entry + attrName + ': ' + value + '\n'

        return entry
```

Umlaute ersetzen

row ist eine Zeile in der CSV-datei z.B:

```
Rimmelspacher,Michael,w,10.04.88,Wasserburg,TH,Neufahrner Str. 7,82031,Muenchen,02283-67794984,1574819974
```

```
def replaceUmlauts(row):
    return list(map(lambda s: s.replace(u'ä', 'ae')
                               .replace(u'ö', 'oe')
                               .replace(u'ü', 'ue')
                               .replace(u'ß', 'ss')
                               , row))
```

Main

CSV parsen

```
def main():
    with open(CSV_FILE, newline='', encoding='latin-1') as f:
        reader = csv.reader(f)
        uidNum = 8000
        firstRow = True

        for row in reader:
            row = replaceUmlauts(row)

            if (firstRow):
                attributes = row
                firstRow = False

            else:
                entry = LdifEntry(uidNum, attributes, row)
                uidNum = uidNum + 1

                fileName = LDAP_DATA_FOLDER + getattr(entry, 'attributes')["Nachname"] + '.ldif'
                file = open(fileName, 'x');
                file.write(str(entry))
```

LDAP - Zugriffsrechte

Anforderung: Ein anonymous bind darf nur die Benutzerkennung erhalten

LDAP - Zugriffsrechte

- Anforderung: Ein anonymous bind darf nur die Benutzerkennung erhalten
- Der OpenLDAP Server auf Ubuntu wird durch den cn=config tree definiert
- Anzeigen der aktuellen Zugriffsrechte mit einer ldapsearch auf das **olcAccess** Attribut:

```
root@vmpsateam09-09:~# ldapsearch -Y EXTERNAL -H ldapi:/// -b cn=config 'olcDatabase={1}mdb'
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth

dn: olcDatabase={1}mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: {1}mdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=team09,dc=psa,dc=in,dc=tum,dc=de
olcAccess: {0}to attrs=userPassword by self write by anonymous auth by * none
olcAccess: {1}to attrs=shadowLastChange by self write by users read
olcAccess: {2}to attrs=uid,entry by anonymous read by * break
olcAccess: {3}to * by self write by anonymous none by users read
olcLastMod: TRUE
olcRootDN: cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de
olcRootPW: {SSHA}Fm+IDJ3HPqNC6Rwzo5fxguYiP3B8FtiE
```

LDAP - Zugriffsrechte

- Erzeugen einer ldif Datei um Zugriffsrechte anzupassen

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to attrs=userPassword
by self write
by anonymous auth
by * none

- add: olcAccess
  olcAccess: {1}to attrs=shadowLastChange
  by self write
  by users read
- add: olcAccess
  olcAccess: {2}to attrs=uid,entry
  by anonymous read
  by * break
- add: olcAccess
  olcAccess: {3}to *
  by self write
  by anonymous none
  by users read
```

LDAP - Zugriffsrechte

- Diese ldif Datei kann mittels folgendem Befehl eingespielt werden:

```
ldapmodify -H ldapi:/// -f access.ldif -D "cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de" -W
```

- Testen

```
ldapsearch -x -h vm09.psa-team09.in.tum.de -b dc=team09,dc=psa,dc=in,dc=tum,dc=de "(uid=\*)"
```

- Konsequenzen: Eigene VM Benutzer Accounts

Erzeugen eines X.509 Zertifikats

Erzeugen eines X.509 Zertifikats

Generieren

```
openssl genrsa 2048 > private.key
openssl req -new -x509 -nodes -sha1 -days 1000 -key private.key > output.cer
```

Konvertieren

```
openssl x509 -outform DER -in output.cer -out binary.der
```

ldif Syntax

```
usercertificate;binary:< file:///$PATH_TO_BINARY_FILE$/outcert.der
```

Erzeugen eines X.509 Zertifikats

Testen

```
root@vmpsateam09-05:~# ldapsearch -x -h 192.168.9.9 -b dc=team09,dc=psa,dc=in,dc=tum,dc=de \
 -D "cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de" -W \
 "(Matrikelnummer=1813607693)" userCertificate
Enter LDAP Password:
# extended LDIF
# LDAPv3
# base <dc=team09,dc=psa,dc=in,dc=tum,dc=de> with scope subtree
# filter: (Matrikelnummer=1813607693)
# requesting: userCertificate
# 1813607693, psaou, team09.psa.in.tum.de
dn: Matrikelnummer=1813607693,ou=psaou,dc=team09,dc=psa,dc=in,dc=tum,dc=de
userCertificate;binary:: MIID5TCCAs2gAwIBAgIUd5HM011N05eAf/WxRUJNQrw+sFIwDQYJK
 oZIhvcNAQEFBQAwgYExCzAJBgNVBAYTAkRFMQowCAYDVQQIDAEtMQ8wDQYDVQQHDAZUZXJsYW4xCj
 ...
 Pu9yOpe+ogRhIMEwngzAXipGjyEyUJfLvp3E86knMNjr/xCXqKU/XEdbv8xbPtfWHmgHWIgToTPBl
 5G1uOnK7EMDiYfMuUS+oKw4+kPCVAYEhALTcOarNkFyjB7qamw=

# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
```

LDAP - Hinzufügen aller ldif Dateien

Zusammenführung der einzelnen Schritte

1. Erzeugen eines X.509 Zertifikat in binary Form für jede Kennung
2. Erzeugen der einzelnen ldif Dateien für jede Kennung
3. Hinzufügen zum LDAP-Server

LDAP - Hinzufügen aller ldif Dateien

Bash Skript addCSVtoLDAP.sh

```
BASE_DIR=/root/workspace/csv2ldif
INPUT_DIR=$BASE_DIR/testdata
CSV_INPUT=$INPUT_DIR/benutzerdaten.csv2
BIN_DIR=$INPUT_DIR/public
KEY_DIR=$INPUT_DIR/private
PWD_FILE=$BASE_DIR/.pw

echo Cleanup

/bin/rm -f $BIN_DIR/*.der
/bin/rm -f $BIN_DIR/*.cer
/bin/rm -f $INPUT_DIR/input.*
/bin/rm -f $KEY_DIR/*.key
/bin/rm -f $BASE_DIR/ldap_data/*.ldif
```

LDAP - Hinzufügen aller ldif Dateien

Bash Skript **addCSVtoLDAP.sh**

```
echo Create Certifcates

cd $INPUT_DIR
export IFS=,; cat $CSV_INPUT |  while read na vn x1 x2 x3 co x5 x6 ci x7 x8; do
   [ $co == "D" ] && co=DE;
   openssl genrsa 2048 > $KEY_DIR/$x8.key
   printf "%s\n-\n%s\n-\n-\n%s %s\n%s.%s@web.de\n"  "DE" "$ci" "$vn" "$na" "$vn" "$na" > $INPUT_DIR/input.$x8;
   cat input.$x8 |  openssl req -new -x509 -nodes -sha1 -days 1000 -key $KEY_DIR/$x8.key > $BIN_DIR/$x8.cer;
   openssl x509 -outform DER -in $BIN_DIR/$x8.cer -out $BIN_DIR/$x8.der ;
done
```

LDAP - Hinzufügen aller ldif Dateien

Bash Skript **addCSVtoLDAP.sh**

```
echo Create ldifs

cd $BASE_DIR
./csv2ldif
```

LDAP - Hinzufügen aller ldif Dateien

Bash Skript **addCSVtoLDAP.sh**

```
echo Import ldifs

cd $BASE_DIR/ldap_data
for i in *.ldif; do
  echo -n "-- add $i "
  ldapadd -x -D "cn=admin,dc=team09,dc=psa,dc=in,dc=tum,dc=de" -f $i -y $PWD_FILE > /dev/null 2>&1
  ret=$?
  [ $ret -eq 0 ] && echo "ok"
  [ $ret -ne 0 ] && echo "error (ret=$ret)"
done
```

SSSD - Installation/Konfiguration

Der System Security Services Daemon ist eine Sammlung von Diensten, die zur Authentifizierung und Sicherheit dienen.

SSSD - Installation/Konfiguration

```
sudo apt install sssd-ldap ldap-utils
```

Änderungen bei der Installation

```
/etc/pam.d/common-*
/etc/nswitch.conf
```

Beispiel `common-auth`

```
auth    [success=1 default=ignore]      pam_sss.so use_first_pass
```

Beispiel `nswitch.conf`

```
passwd:         files systemd sss
```

SSSD - Installation/Konfiguration

Anlegen einer `/etc/sssd/sssd.conf`

```
[sssd]
config_file_version = 2
domains = psa-team09.in.tum.de

[domain/psa-team09.in.tum.de]
id_provider = ldap                                  # use LDAP for id resolution
auth_provider = ldap                                # use LDAP for authentification
ldap_uri = ldap://vmpsateam09-09.psa-team09.in.tum.de   # verbindung zum ldap-server
cache_credentials = True
ldap_search_base = dc=team09,dc=psa,dc=in,dc=tum,dc=de  # base domain des ldap-servers
ldap_id_use_start_tls = true                        # use TLS connection
ldap_default_bind_dn = cn=vm05,ou=computers,dc=team09,dc=psa,dc=in,dc=tum,dc=de   # account für bind an den ldap server
ldap_default_authtok_type = password                # art der authentifikation am ldap-server
ldap_default_authtok = XXXXXXXXX                    # passwort für ldap-server account
ldap_tls_reqcert = allow
```

SSSD - Installation/Konfiguration

- Starten des sssd Services:

```
sudo systemctl start sssd.service
```

- Aktivieren der automatischen Erzeugung von home directorys - nutzen des im LDAP server hinterlegten home Verzeichnis-Pfad:

```
sudo pam-auth-update --enable mkhomedir
```

- Testen

```
root@vmpsateam09-04:~# ldapwhoami -x -ZZ -h vmpsateam09-09.psa-team09.in.tum.de
anonymous
```

SSSD - Installation/Konfiguration

- Löschen der lokalen Nutzer Einträge

```
userdel nutzerkennung # ohne löschen des homeverzeichnisses

# oder manuell aus den beiden lokalen dateien löschen

#/etc/passwd
#/etc/shadow
```

- Testen

```
id -a userkennung
su userkennung
passwd # als user
```

Anmerkungen

- slapd debug

```
debug kurz: slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /etc/ldap/slapd.d -d 256
debug lang : slapd -h "ldap:/// ldapi:///" -g openldap -u openldap -F /etc/ldap/slapd.d -d 1023
```

- sssd Cache leeren

```
sss_cache -E
systemctl restart sssd.service
```

Fragen?

Quellen