

# Network Engineering Assignment 2

---

WEBSITE LDAP AUTHENTICATION

I give permission to have this assignment shared for peer evaluation.

## 1 EVALUATE LDAP SERVER OPTIONS

---

### 1.1 OPENLDAP

OpenLDAP is the most common LDAP server, and with that it has the most amount of help online, whether that be through commercial support or questions & answers on forums. All modifications to the LDAP server are done through the CLI. [1], [2]

**Licensed:** OpenLdap Public Licence

### 1.2 APACHE DIRECTORY SERVER

Often shortened to ApacheDS, this is Apache's implementation of a LDAP directory server. It has some notable advantages over OpenLDAP, them being. One ApacheDS is available for all the major OS's and two Apache provides an application called the Apache Directory Studio, which makes managing the LDAP server much easier with a proper UI. [1], [2]

**Licensed:** Apache License

### 1.3 389 DIRECTORY SERVER

Although developed by Red hat, 389 Directory server is available for all linux distros. It's high performance and much like ApacheDS comes with a GUI that makes managing the LDAP server much easier for humans. [1], [2]

**Licensed:** GPL

### 1.4 AZURE AD

An alternative to using LDAP while sticking to an active directory is Microsoft Azure AD. It's the most common active directory system used in the corporate world due to the prevalence of Microsoft and the ease of management, however it does not support LDAP but instead uses REST API's, as well it's not open sourced. This isn't really suited for this assignment due to the lack of LDAP support, however it is included to showcase a similar alternative to the LDAP servers. [3]

## 2 INSTALL PROCESS

---

openLDAP was chosen to be the LDAP server for this documents installations purposes

### **Dependencies:**

The only main dependency that is needed is **Cyrus-SASL2**, however the fresh ubuntu install I used already had this installed. This is not a mandatory dependency but will be used for SSL/TLS with the LDAP server.

## 2.1 INSTALL & CONFIGURE LDAP

The first step is to install and setup the ldap server, detailing how to configure the ldap server is not in the specifications of this assignment however, so it will not be included but follow the pop ups after running the second command below.

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

After configuration, you will need to populate the LDAP server, previous work done in Workshop unit 2 was used for this and explaining how LDIF files work is not in the specifications of this assignment.

Below is the command to add to LDAP, unit2-solution.ldif can be replaced with any LDIF file.

```
ldapadd -x -D "cn=admin,dc=localhost" -W -f unit2-solution.ldif
```

## 2.2 SETUP AUTHENTICATION VIA LDAP

NGINX & Wiki.js were the Server & Wiki chosen for Assignment 1 and will still be used here.

Wiki.js provides LDAP authentication support, to enable this edit the Wiki.js Config file using any text editor, I use nano.

The below command also assumes that the wiki.js config was placed in that location, as was described in Assignment 1.

```
sudo nano /var/www/wiki.js/config.yml
```

Scroll down until you see the LDAP section under authentication and change look at the screenshot below to see how to properly enable LDAP authentication.

```
GNU nano 2.9.3 /var/www/wiki.js/config.yml

  clientId: GITHUB_CLIENT_ID
  clientSecret: GITHUB_CLIENT_SECRET
slack:
  enabled: false
  clientId: SLACK_CLIENT_ID
  clientSecret: SLACK_CLIENT_SECRET
ldap:
  enabled: true
  url: 'ldap://localhost:389'
  bindDn: cn=admin,dc=localhost
  bindCredentials: toor
  searchBase: dc=localhost
  searchFilter: '(uid={{username}})'
  tlsEnabled: true
  tlsCertPath: '/etc/ldap/sasl2/ca-certificates.crt'
azure:
  enabled: false
  clientId: APP_ID
  clientSecret: APP_SECRET_KEY

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

The important variables that may need to be changed for each LDAP server are:

**bindDN:** The main root user for LDAP when accessing the LDAP server.

**bindCredentials:** The password for accessing the LDAP server as the main root user.

**searchBase:** the base LDAP directory to start searching for users

**tlsEnabled:** This can remain false for now but will be enabled later in this Document, as this is for the SSL/TLS section.

## 2.3 USERS & TESTING

As stated before I just used the Workshop 2 LDAP files for creating organizations & persons. But below I have included an example of what a person would need to be given in said LDIF file to be able to connect to the wiki.









```

1  dn: cn=Qwerty test,ou=Pest control,dc=Conglomerated
   Exports,dc=localhost
2  objectclass: organizationalPerson
3  objectclass: inetOrgPerson
4  cn: Qwerty test
5  sn: test
6  mail: test.man3@email.com.au
7  userPassword: qwerty123
8  uid: qwerty

```

Where uid is the username, and userPassword is the password to login to the wiki. Also a note is that wiki.js documentation says users will need an email address as well to be able to login.

Below is just a screenshot showing the users in the wiki, just as proof that it does authenticate to the LDAP server. Qwerty test can login

Users 					
Manage users and access rights					
	Name	Email Address	Provider	Created On	Updated On
	Administrator	admin@example.com	 Local	Jun 26, 2019 4:48 PM	Jun 26, 2019 4:48 PM
	Guest	guest	 Local	Jun 25, 2019 3:31 PM	Jun 25, 2019 3:31 PM
	Qwerty test	test.man3@email.com.au	LDAP / Active Directory	Jun 26, 2019 5:10 PM	Jun 27, 2019 4:56 PM
	aaa	aaa@aaa.com	 Local	Jun 26, 2019 5:05 PM	Jun 26, 2019 5:05 PM

## 2.4 CREATE SSL KEY

Firstly a SSL key will need to be created, follow below.

```

sudo -i
cd /etc/ssl/private
openssl genrsa -aes128 -out ldap_ssl.key 4096
enter passphrase: (mine was toor)

```

```

sam@server:/etc/ssl$ sudo -i
root@server:~# cd /etc/ssl/private
root@server:/etc/ssl/private# openssl genrsa -aes128 -out ldap_ssl.key 4096
Generating RSA private key, 4096 bit long modulus
.....++
.....
++
e is 65537 (0x010001)
Enter pass phrase for ldap_ssl.key:
Verifying - Enter pass phrase for ldap_ssl.key:
root@server:/etc/ssl/private#

```

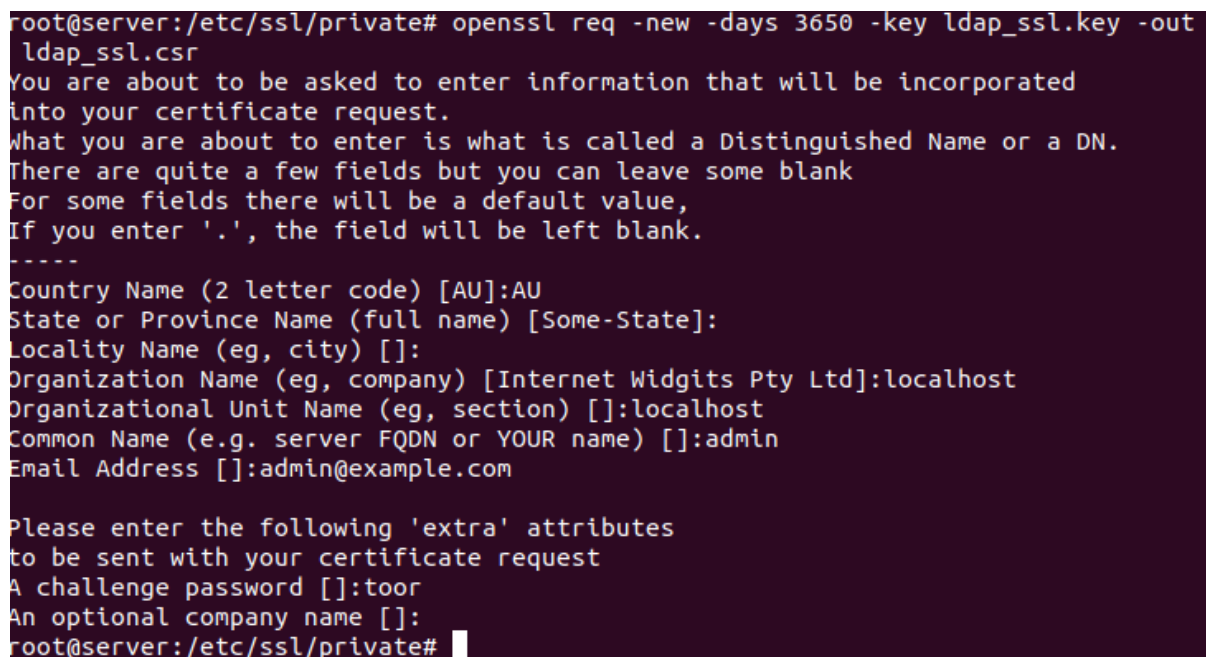
Once the key has been created, the passphrase that was just used when creating the key will need to be removed, SSL doesn't let you create a key without a passphrase, and LDAP will need the key removed.

```
openssl rsa -in ldap_ssl.key -out ldap_ssl.key
```

Next a CSR or certificate signing request will need to be generated and given to the key, to start the generating process type the command below.

```
openssl req -new -days 3650 -key ldap_ssl.key -out ldap_ssl.csr
```

This will ask a few questions that will be incorporated into the certificate request, these are pretty straight forward questions, however I have included the screenshot of what I used.

A terminal window showing the execution of the 'openssl req -new -days 3650 -key ldap\_ssl.key -out ldap\_ssl.csr' command. The prompt is 'root@server:/etc/ssl/private#'. The output shows a series of prompts for certificate information: Country Name (2 letter code) [AU]:AU, State or Province Name (full name) [Some-State]:, Locality Name (eg, city) [], Organization Name (eg, company) [Internet Widgits Pty Ltd]:localhost, Organizational Unit Name (eg, section) [], Common Name (e.g. server FQDN or YOUR name) []:admin, and Email Address []:admin@example.com. After these, it asks for 'extra' attributes: A challenge password []:toor and An optional company name []. The final prompt is 'root@server:/etc/ssl/private#' with a cursor.

```
root@server:/etc/ssl/private# openssl req -new -days 3650 -key ldap_ssl.key -out
ldap_ssl.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AU
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:localhost
Organizational Unit Name (eg, section) []:localhost
Common Name (e.g. server FQDN or YOUR name) []:admin
Email Address []:admin@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:toor
An optional company name []:
root@server:/etc/ssl/private#
```

Then sign the key

```
openssl x509 -in ldap_ssl.csr -out ldap_ssl.crt -req -signkey ldap_ssl.key -days 3650
```

## 2.5 ENABLE SSL FOR LDAP

Firstly you need to copy the key/certificate to the ldap directory, /etc/ldap/sasl2/

```
cp /etc/ssl/private/{ldap_ssl.key,ldap_ssl.crt} /etc/ssl/certs/ca-certificates.crt /etc/ldap/sasl2/
```

Next to enable SSL for LDAP, a LDIF file is used to modify the server, just create a LDIF file anywhere and call it anything, I just used ldap\_ssl.

```
nano ldap_ssl.ldif
```

And enter what is in the screenshot below, renaming the key/certificate files to whatever you called them.

```
GNU nano 2.9.3      ldap_ssl.ldif
dn: cn=config
changetype: modify
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ldap/sasl2/ca-certificates.crt
-
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ldap/sasl2/ldap_ssl.crt
-
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ldap/sasl2/ldap_ssl.key
```

Now use `ldapmodify` with this file to actually modify the LDAP server

```
ldapmodify -Y EXTERNAL -h ldapi:/// -d ldap_ssl.ldif
```

If all goes well then all that's left now is to go back into the `wiki.js` config file where we enabled LDAP authentication before and enable TLS and set the path to the certificate.

```
sudo nano /var/www/wiki.js/config.yml
```

```
ldap:
  enabled: true
  url: 'ldap://localhost:389'
  bindDn: cn=admin,dc=localhost
  bindCredentials: toor
  searchBase: dc=localhost
  searchFilter: '(uid={{username}})'
  tlsEnabled: true
  tlsCertPath: '/etc/ldap/sasl2/ca-certificates.crt'
```

### 3 BIBLIOGRAPHY

---

- [1] R. Bhargava, "Choosing an LDAP Server," *JumpCloud*, 06-Mar-2019. .
- [2] E. Stani, "Top 4 open source LDAP implementations," *opensource*, 18-May-2014. .
- [3] G. Cowser, "AD DS vs Azure AD – So what's the difference?," *New Horizons Australia*, 26-Jan-2017. .