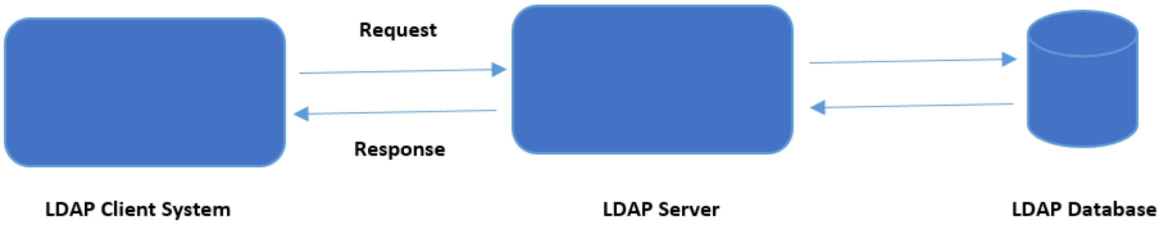
LDAP Authentication Using Node JS

Lightweight Directory Access Protocol (LDAP) is a protocol which helps to maintain distributed directory information in an easy manner. It allows to store a directory of items and information about the directories. In real time, several directory servers are available for storing data in directory format. These directories are a set of objects with data organized in different ways such as logical and hierarchical. LDAP is used to store information in the tree structure format. It was developed as a replacement for Directory Access Protocol. The key purpose of LDAP is to provide centralized authentication, which stores details such as usernames and passwords in the directory. It supports various applications or services for valid users with the help of plug-ins. There are many LDAP servers available for LDAP authentication. Organizations develop diverse servers based on their user and business usage perspectives. Few examples include Microsoft Azure Active Directory, Apache Active Directory and Linux Server Active Directory.

LDAP is a mechanism for data security in organizations, where the data is stored in hierarchical or tree format. There are different LDAP servers available to store the data for different organizations. This data is stored with attributes such as cn, sn, o, ou, dn and dc. Here cn denotes the common name, sn indicates the surname and dc represents domain component. They provide authentication and authorization to the application. For LDAP authentication, we need access directory, which stores data in the tree structure based on the above attributes.

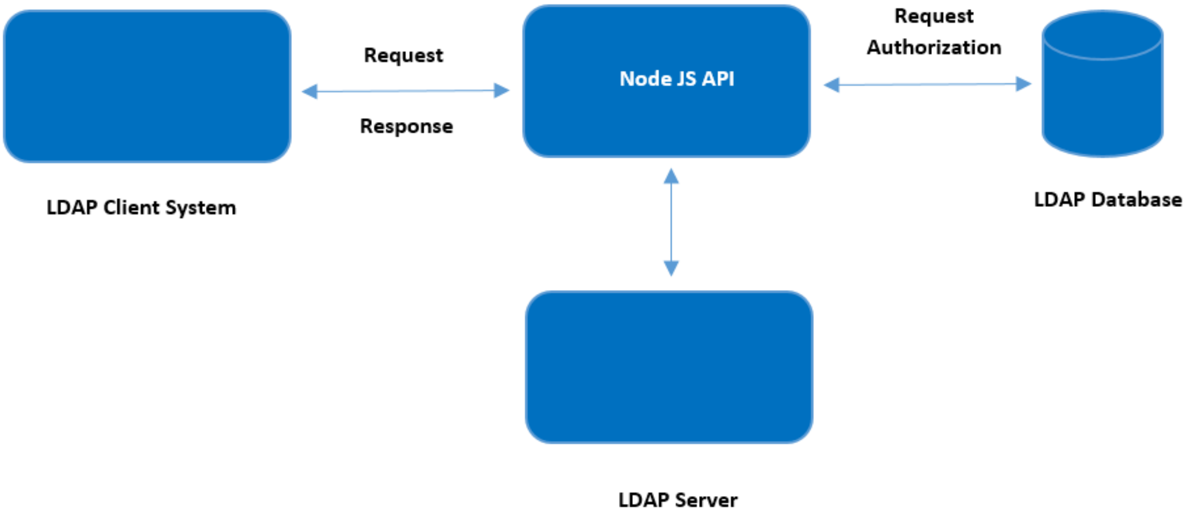
How does LDAP work?

LDAP not only stores usernames and passwords but also stores various attributes. LDAP authentication follows a Client-Server architecture, where the client requests information from LDAP database and the server is the LDAP server. Based on the client’s request, the server communicates with LDAP database for authentication and upon successful authentication, it sends the requested data in response format to the client.



How is LDAP used in Active Directory for Application or Project?

Different organizations provide different servers for using business applications. LDAP offers data security for real time applications or projects. The following procedure is used to implement LDAP in an application. Let us see how we use Apache Directory Studio for LDAP server, to integrate Node JS application.



- Install Apache Directory Studio based on the operating system.
- After starting the server you can see the LDAP browser on left side, which contains Directory Information Tree (DIT) which in turn contains Active Directories (ADs) as well as Root Entry which contains children.
- Go to ou: system, select ou: users and add the users in AD.
- DIT contains two object classes - Structural and Auxiliary, which form the basis for entry creation. There are several object classes available in the directory such as Country, Organization, Organization Unit, inetOrgPerson, Group of Names and so on.
- Based on the above object classes we create entries and select the required object classes.
- After the selection of object classes, we proceed to select the parent class for the objects. To do this, we select the Relative Distinguished Name (RDN) and give an equivalent name for that RDN. For example, we can select cn from RDN and give the name as raj.
- After this, Distinguished Name (DN) will be automatically displayed based on Parent and RDN. Now, click on Next and provide any further missing attributes’ values and click on Finish. This enables the user to be added.

- Give the authentication for the details - add uuid and password attributes and provide values for the attributes. Once we enter the password, it will be converted using salt (hex) mechanism.
- We can use different APIs for LDAP using Node JS. The following procedure connects LDAP using Node JS:

i. Install ldapjs library using the following command in Node JS application:

```
npm install ldapjs
```

ii. Create the client for LDAP server using the following syntax:

```
var client =ldap.createClient(username, password)({
url: “ldap:// local host: port”
})
```

iii. Communicate with LDAP server using the credentials with Bind. The syntax is as follows:

```
client.bind(username, password,  function(error){
if(error){
}
else{
}
```

We can thus establish binding to LDAP server, make use of the diverse range of operations such as Add user, Get All, Update User and so on and obtain data from AD using Node JS.

Advantages

- ✓ LDAP is simple and provides an enhanced level of security to the application.
- ✓ It supports different types of applications, we can integrate it with different applications.
- ✓ It is used to perform simple operations like insert, update, delete and retrieve on AD.

Conclusion

LDAP authentication provides security for user details that are stored in the AD in LDAP server. The account details are securely stored in AD and we can get them using the Search service, modify them and even delete them if required. We can use various client side APIs to perform a wide range of operations on LDAP using AD.

Contact for further details

Threenatha Reddy K R

Associate Software Engineer - Emerging Technologies

threenathak.in@mouritech.com

MOURI Tech