LDAP authentication best practices

Revision 20210304

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Goal

This article describes the best practices to connect to VDP databases using LDAP authentication.

Content

The recommended steps to connect to a VDP database using LDAP authentication are:

1) Define a LDAP data source to a LDAP or Active Directory server.
2) Create a database with LDAP as the Authentication type or enable LDAP Authentication for the Virtual DataPort Server.
3) Create roles of the LDAP server.
4) Assign privileges to the LDAP roles to connect to the LDAP database.

1) Define an LDAP data source to an LDAP or Active Directory server

To create an LDAP data source the **Server URI** and the credentials (**Login** and **Password**) to access the LDAP data source have to be provided. (section LDAP Sources in the Virtual DataPort Administration Guide)

The URL and credentials for the LDAP data source are the same that can be used from any third party LDAP client and such clients can be used to test the connections to the LDAP server. The expected format of the connection URI is `ldap://<host_name>:<port>/`. Optionally, the base Distinguished Name (DN) can also be appended to the base URI.
In case the LDAP users are organized in Domains the domain name has to be provided as part of the login information using the syntax `<domain>\<username>`.

2) Create a database with LDAP as the Authentication type or enable LDAP Authentication for the Virtual DataPort Server

The next step is to create either a database with LDAP as Authentication type or, since Denodo 8.0, to enable the global authentication for the whole Virtual DataPort Server which can be done under `Administration > Server configuration > Server authentication` via the Virtual DataPort Administration Tool. Creating a database with enabled LDAP Authentication can be done under `Database Management > New > Authentication: LDAP Authentication`.

For either of these options - enabling global LDAP Authentication or creating an LDAP authenticated database - it will be necessary to specify the user and role search patterns. In order to find the right pattern, we recommend the use of any third party LDAP client, like JXplorer, in order to execute and test the queries to the LDAP server if needed.
The information that is required from the LDAP administrator is:

- **User base**: node of the LDAP server that is used as scope to search nodes that represent users. It is possible to specify more than one. For instance: `CN=Users,DC=acme,DC=loc`.
- **Attribute with user name**: name of the attribute that will be used as login identifier. For example: `CN`

- **User search pattern**: pattern used to retrieve the users from the LDAP. For example: `(&(objectClass=person))`
- **Role base**: node of the LDAP server that is used as scope to search nodes that represent users. It is possible to specify more than one. In some examples is the same that the **User base**. for instance: `CN=Roles,DC=acme,DC=loc`
- **Attribute with role name**: name of the attribute that contains the name of the role. For example: **CN**
- **Role search pattern**: pattern used to generate the LDAP queries that will be executed to obtain the nodes that represent the roles of a user. This pattern has to contain the token `@{USERDN}`, which will be replaced with the Distinguished Name of the user that tries to connect to the database. For example: `(&(member=@{USERDN})(objectClass=group))`.
  In this example, we are looking for all the objects with the object class group where the member is the Distinguished Name. For example if the user `jsmith` was trying to connect the Distinguished Name for this user will be:
  ```
        CN=jsmith,CN=Users,DC=acme,DC=loc
  ```
  and the example search for the roles will be:
  ```
        (&(member=CN=jsmith,CN=Users,DC=acme,DC=loc)
        (objectClass=group))
  ```

3) Create roles of the LDAP server

This can be done manually or with the import tool.

To import a big number of roles from an LDAP it is recommended to use the **Import Roles tool**. (section Creating Roles of the Virtual DataPort Administration Guide).
If the role `serveradmin` is imported from the LDAP server it will be ignored for security reasons. To assign global administrator privileges to an user it will be necessary to assign to some of the other roles returned by the LDAP server the `serveradmin` role.

To import just a subset of all the existing roles in an LDAP server it is recommended to manually create the roles instead of iterating on the list of roles for example using the VDP Admin tool to add the roles one by one or to use a VQL script to create the roles using the VQL Shell. To create a new role using VQL the following statement can be executed:

```
CREATE ROLE <role_name> '<role_description>';
```

4) Assign the privileges to the LDAP roles to connect to the LDAP database

To assign privileges to a role it is recommended to use several VDP admin tool sessions simultaneously to simplify the process:

a) a VDP Admin tool connected with an admin user that configures the privileges for the role .
b) a VDP Admin tool connected with a normal user that belongs to the role whose privileges are being assigned.

When connecting with a normal user, in case of error, the recommendation is to open the `vdp.log` file under the `<DENODO_HOME>\logs\vdp` folder. If there is not enough information about the error it is possible to set up the log category to **trace** using the **logcontroller** stored procedure. For example, the following commands:

```
call logcontroller('com.denodo.vdb.security.LDAPDatabaseAuthenticator', 'trace');
call logcontroller('com.denodo.vdb.catalog.user.User', 'trace');
```

```
call        logcontroller('com.denodo.vdb.security.DefaultLDAPUserGroupRetriever',
'trace');
call
logcontroller('com.denodo.vdb.engine.wrapper.raw.ldap.executor.LDAPExecutor',
'trace');
call logcontroller('com.denodo.vdb.catalog.role.Role', 'trace');
```

will enable trace information in the logs for the categories relevant to the LDAP authentication. It is important to remember to change these categories back to error once the problem has been diagnosed and solved.

In addition to these technical considerations about LDAP and user authentication, there are some aspects to take into account regarding the connections. When using the LDAP authentication to connect to a database, a latency for the authentication is added.
The recommendation to avoid a big overload due to the LDAP authentication is to configure the clients to use Connection Pools against VDP. That will ensure that not all the queries need to create the connection and authenticate against the LDAP server.
For example, using JDBC access it is possible to use a connection pool to avoid the overhead.

References

Virtual DataPort Administration Guide: LDAP Sources.
Virtual DataPort Administration Guide: LDAP Authentication
Virtual DataPort Administration Guide: Roles.
Virtual DataPort Administration Guide: Administration of Databases, Users, Roles and their Access Rights.
LDAP Authentication at server level
Importing LDAP roles in Virtual DataPort