High scalability and availability of AIX secldapclntd using the Tivoli Directory Server proxy

Skill Level: Intermediate

Nikhil Firke (nikhilfirke@in.ibm.com)
System Software Engineer
IBM India Software Labs

Nilesh T. Patel (nilesh.patel@in.ibm.com)
System Software Engineer
IBM India Software Labs

01 Sep 2009

The secldapclntd daemon provides and manages connection between the AIX® security LDAP load module of the local host and an LDAP server, and handles transactions from the LDAP load module to the LDAP server. Simple configuration steps do not allow us to specify highly available and scalable LDAP servers at the back end. This article lists the steps to configure a highly available and scalable back-end LDAP for the secldapclntd daemon using the Tivoli® Directory Server proxy.

Introduction

In any distributed environment, in order to maintain universal access, consistent authentication and authorization services are a necessity. A large number of such deployments use the IBM Directory Server (TDS) for the purpose of centralized security.

Centralizing the authentication and password services not only increases security but also reduces the administrative overhead associated with it. A good solution to this problem is to have a single server that handles all the requests in an environment. However, this advantage comes at the cost of high availability of the server. High availability is always desired in any user authentication solution, as it might cause the entire enterprise to stop working if the single authenticating server...
goes down. With LDAP, there are two options to combat this situation. IBM Tivoli Directory Server provides the replication features that can be used to configure servers in a master-master mode or in a master-replica mode. In the master-master configuration, there is no primary server and the changes made to any of the server get propagated to the other masters. In the master-replica setup, all changes occur on the master server and the updates are propagated to one or more replicas. The replication can be configured to occur on a scheduled basis or immediately after the change is completed on the master server.

To demonstrate a scalable solution, this article introduces a TDS proxy in the secldapclntd configuration. A large amount of user information defined in a single LDAP server can cause the response time to increase and the overall system may slow down. To increase the scalability of the system, this article proposes a solution with distributed directories. A distributed directory is a directory environment in which data is partitioned across multiple directory servers. Proxy server is a special type of IBM Tivoli Directory server which is capable of receiving requests from the clients and routing them as required. Apart from routing the requests, the proxy server is also capable of load balancing, fail over, and distributed authentication. Since the clients in the environment are only aware of the front-end proxy server, the server cluster behind the proxy server remains hidden and a unified directory view is presented to the client.

Introduction to TDS proxy server

LDAP is used to store any kind of directory-like information that requires fast lookups and less-frequent updates. Enterprise-wide LDAP servers are expected to store millions of entries. To support such a huge number of entries and the requests corresponding to them and yet not have any degraded performance calls for highly scalable hardware. To address these issues, a distributed directory can be used.

Distributed directories were introduced to address performance and scalability issues. Distributed directories help in dividing the huge amount of data over more than one server. As the data gets distributed, the number of requests hitting any particular server is automatically reduced, making the setup easier to manage.

This distributed directory setup is very effectively abstracted by adding a proxy server in front of it. Clients know only about the proxy and the proxy server is configured with the details of distributed data at the back end. The proxy server is configured with the methodology of splitting the data. This allows the proxy to get the data from the back end and transfer it to the clients. The interaction between the proxy server and the back-end directory servers is entirely transparent to the clients. A proxy server can also act as a load balancer or a fail-over manager.

The key features of Tivoli Directory proxy server are:
**Scalability**: Scalability is an important requirement from a directory server. Distributed Directory effectively scales any directory setup.

There are different ways of setting up a distributed directory. The currently available mechanisms are:

- **RDN Hash**-based splitting: The directory server entry is used to calculate a unique hash value. Hash value is calculated on the basis of the RDN of the entry. When the directories are distributed, this hash value maps to a specific entry in a specific directory server in the back end.

- **Subtree**-based splitting: In this splitting mechanism, each subtree is configured to reside on a separate directory server in the back end.

- **Abstraction**: The proxy server acts as a layer of abstraction over a set of directory servers. A client views the complete topology as a unified directory and has no knowledge of the distribution at the back end. Also the interaction between the back end and the directory server is transparent to the clients.

### Set up secldapclntd with the TDS proxy

The mksecldap command can be used to set up the client and the server for any typical setup with secldapclntd and IBM Tivoli Directory Server. At a very high level, the following steps are performed by the mksecldap command while configuring the server:

2. Configures the suffix under which the AIX users and the groups will be stored.
3. The LDAP database is loaded with the user and group information from the localhost's security database files.
4. The LDAP server administrator DN and password is configured.
5. AIX audit plug-in for the LDAP server is installed.
6. LDAP server is started.
7. The default password encryption mechanism is changed to crypt.
8. Adds the LDAP server entry (slapd) to /etc/inittab for automatic restart after reboot.
This article does not use the mksecldap command to set up the LDAP server. We will be avoiding using mksecldap because our final aim is to configure the AIX clients with the TDS proxy server and not the TDS RDBM back end. If we use the mksecldap command, we will end up having a TDS RDBM server configured with the AIX clients. TDS RDBM server has a DB2® back end, which we do not need.

We need a TDS proxy server configured with the AIX clients. Proxy servers do not have a DB2 back end.

All the AIX data is present under the cn=aixdata subtree. This subtree, in turn, has other separate subtrees under it that hold the user entries, group entries, and other necessary information. User information is present under ou=people,cn=aixdata. Since the subtree ou=people, cn=aixdata, takes the maximum number of hits, we will be splitting this subtree over two TDS back-end servers such that the load gets distributed over two back ends behind the proxy. The top-level subtree base cn=aixdata will be on only one back-end server behind the proxy.

We are going to set up an environment, which looks like the following:

**Figure 1. Sample configuration**

The figure details are:

1. TDS proxy: TDS proxy is the Tivoli Directory Proxy Server, which the
clients in the environment will be aware of.

2. Blue lines: Blue lined behind the proxy server show the replication topology for cn=aixdata and cn=ibmpolicies.

3. Black lines: Black lines behind the proxy server show the replication topology for ou=People, cn=aixdata.

4. secldapinst1a and secldapinst1b: These are the replicating RDBM splits that contain the complete subtree for cn=aixdata and cn=ibmpolicies and partial subtree for ou=People, cn=aixdata.

5. secldapinst2a and secldapinst2b: These are the replicating RDBM splits that contain the partial subtree for ou=People, cn=aixdata. Complete subtree for cn=aixdata (other than the complete subtree for ou=People, cn=aixdata) and cn=ibmpolicies would be present here because of the replication topology.

6. Server Group 1: Server Group 1 groups secldapinst1a and secldapinst1b together, making it highly available.

7. Server Group 2: Server Group 2 groups secldapinst2a and secldapinst2b together, making it highly available.

Steps to set up the replication for back ends

We will be basically concentrating on the data under the cn=aixdata and cn=ibmpolicies. Data under cn=aixdata (except the subtree ou=people, cn=aixdata) and cn=ibmpolicies will be present across all the four servers in our configuration. Before we start setting up the replication, following basic steps are required to create the TDS instance and get them ready for replication configuration.

- Create four TDS instances using the following commands, and run these commands to create and configure ITDS RDBM instance:

Listing 1. Create and configure TDS RDBM instances

Add user aixauth
# idsadduser -g idsldap -l /home/aixauth -u aixauth -w aixauth

Create TDS instance
# idsicrt -I aixauth -e abcd1234567890 -l /home/aixauth -n

Configure database instance with this TDS instance
# idsfcfgdb -I aixauth -l /home/aixauth -a aixauth -w aixauth -t aixauth -n
Configure Admin username and password
# idsdnpw -I aixauth -u cn=root -p root -n

Configure cn=aixdata suffix
# idscfgsuf -I aixauth -s cn=aixdata -n

Start TDS instance to change default password encryption
# ibmslapd -I aixauth -n
<

Change default password encryption to crypt
# idsldapadd -D cn=root -w root
dn: cn=Configuration
changetype : modify
replace : ibm-slapdPwEncryption
ibm-slapdPwEncryption: crypt

Let's assume that the four instances created are on separate machines and each of these instances has been named as aixauth. Here, we are assuming that the hostname for the four boxes is secldapinst1a.ibm.com, secldapinst1b.ibm.com, secldapinst2a.ibm.com and secldapinst2b.ibm.com.

• Set up replication for cn=aixdata and cn=ibmpolicies.
  Set up peer-to-peer replication between the four instance for cn=aixdata subtree. The replication for cn=aixdata subtree should look like this:

Figure 2. Sample configuration
• Set up peer-to-peer replication topology as a group of two instances between secdapinst1a.ibm.com and secdapinst1b.ibm.com (as Server Group 1) and secdapinst2a.ibm.com and secdapinst2b.ibm.com (as Server Group 2) for the subtree ou=people,cn=aixdata.

Setting up of a different replication topology under a base that already has replication set up can be quite tedious. Care must be taken while setting up this topology. In the previous step, we had set up replication for cn=aixdata and now we are configuring different replication topology for the ou=people,cn=aixdata entry, which is under cn=aixdata. As per the TDS design, we can have such a configuration wherein we have nested replication topologies.

The replication for this subtree should look like this:

**Figure 3. Replication topology for ou=people,cn=aixdata**
TDS proxy configuration

- Add Global Admin group member
  Run the following commands on any back-end servers. This needs to be
done on only one instance, because as there is a replication on
cn=ibmpolicies already set up, it will replicate to the other three servers.

Listing 2. Command to add Admin Group member

```
# idsldapadd -h secdapinst1a.ibm.com -D cn=root -w root
dn: cn=manager,cn=ibmpolicies
objectclass: person
sn: manager
cn: manager
userpassword: sec001ret

# idsldapmodify -h secdapinst1a.ibm.com -D cn=root -w root
dn: globalGroupName=GlobalAdminGroup,cn=ibmpolicies
changetype: modify
add: member
member: cn=manager,cn=ibmpolicies
```

- Create and configure a TDS proxy instance.
  Run the following commands to create and configure TDS proxy instance.

Listing 3. Create and configure TDS proxy instance
Add user proxy
# idsadduser -g idsldap -l /home/proxy -u proxy -w proxy

Create TDS proxy instance
# idsicrt -I proxy -e abcd1234567890 -l /home/proxy -x -n

Configure Admin username and password
# idsdnpw -I proxy -u cn=root -p root -n

Start the TDS proxy instance in configuration mode to change default password encryption
# ibmslapd -I proxy -a

Change default password encryption to crypt
# idsldapmodify -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=Configuration
changetype : modify
replace : ibm-slapdPwEncryption
ibm-slapdPwEncryption: crypt

- Define naming context in TDS proxy instance. Run the following commands on TDS proxy instance to add cn=aixdata, cn=ibmpolicies and ou=people,cn=aixdata as naming contexts.

Listing 4. Define the naming contexts in TDS proxy server

# idsldapmodify -h seclldapproxy.ibm.com -D cn=root -w root
dn: cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
changetype : modify
add : ibm-slapdSuffix
ibm-slapdSuffix: cn=aixdata
- add : ibm-slapdSuffix
ibm-slapdSuffix: ou=people, cn=aixdata
add : ibm-slapdSuffix
ibm-slapdSuffix: cn=ibmpolicies

- Define the four back-end servers to the TDS proxy server. Run the following commands on the TDS proxy instance, and it will register all four back ends to the TDS proxy with a connection pool of 10.

Listing 5. Defining the back-end servers

# idsldapadd -h seclldapproxy.ibm.com -D cn=root -w root
dn: cn=seclldapinst1a,cn=ProxyDB,cn=Proxy Backends,cn=IBM Directory,\ cn=Schemas,cn=Configuration
cn: seclldapinst1a
### Define the Server Group 1 and Server Group 2.

Defining the server groups ensures that high availability is maintained. If the proxy server is unable to contact a back-end server, or if authentication fails, then the proxy server startup fails and the proxy server starts in configuration-only mode by default, unless server groupings have been defined in the configuration file. Server groupings enable the user to state that several back-end servers are mirrors of each other, and the proxy server processing can continue even if one or more back-end servers in the group is down, assuming that at least one back-end server is online. Connections are restarted periodically if the connections are closed for some reason, such as the remote server is stopped or restarted.

<table>
<thead>
<tr>
<th>ibm-slapdProxyBindMethod: Simple</th>
<th>ibm-slapdProxyConnectionPoolSize: 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibm-slapdProxyDN: cn=manager,cn=ibmpolicies</td>
<td>ibm-slapdProxyPW: sec001ret</td>
</tr>
<tr>
<td>ibm-slapdProxyTargetURL: ldap://secldapinst1a.ibm.com:389</td>
<td></td>
</tr>
<tr>
<td>objectClass: top</td>
<td></td>
</tr>
<tr>
<td>objectClass: ibm-slapdProxyBackendServer</td>
<td></td>
</tr>
<tr>
<td>objectClass: ibm-slapdConfigEntry</td>
<td></td>
</tr>
</tbody>
</table>

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=secldapinst1b,cn=ProxyDB,cn=Proxy Backends,cn=IBM Directory,\ncn=Schemas,cn=Configuration
cn: secldapinst1b
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 10
ibm-slapdProxyDN: cn=manager,cn=ibmpolicies
ibm-slapdProxyPW: sec001ret
ibm-slapdProxyTargetURL: ldap://secldapinst1b.ibm.com:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=secldapinst2a,cn=ProxyDB,cn=Proxy Backends,cn=IBM Directory,\ncn=Schemas,cn=Configuration
cn: secldapinst2a
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 10
ibm-slapdProxyDN: cn=manager,cn=ibmpolicies
ibm-slapdProxyPW: sec001ret
ibm-slapdProxyTargetURL: ldap://secldapinst2a.ibm.com:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=secldapinst2b,cn=ProxyDB,cn=Proxy Backends,cn=IBM Directory,\ncn=Schemas,cn=Configuration
cn: secldapinst2b
ibm-slapdProxyBindMethod: Simple
ibm-slapdProxyConnectionPoolSize: 10
ibm-slapdProxyDN: cn=manager,cn=ibmpolicies
ibm-slapdProxyPW: sec001ret
ibm-slapdProxyTargetURL: ldap://secldapinst2b.ibm.com:389
objectClass: top
objectClass: ibm-slapdProxyBackendServer
objectClass: ibm-slapdConfigEntry
Listing 6. Defining server groups

```
# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=serverGroupA, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
   cn: serverGroupA
ibm-slapdProxyBackendServerDN: cn=secldapinst1a,cn=ProxyDB,cn=Proxy Backends, cn=IBM Directory, cn=Schemas,cn=Configuration
ibm-slapdProxyBackendServerDN: cn=secldapinst1b,cn=ProxyDB,cn=Proxy Backends, cn=IBM Directory, cn=Schemas,cn=Configuration
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendServerGroup

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=serverGroupB, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
   cn: serverGroupB
ibm-slapdProxyBackendServerDN: cn=secldapinst2a,cn=ProxyDB,cn=Proxy Backends, cn=IBM Directory, cn=Schemas,cn=Configuration
ibm-slapdProxyBackendServerDN: cn=secldapinst2b,cn=ProxyDB,cn=Proxy Backends, cn=IBM Directory, cn=Schemas,cn=Configuration
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendServerGroup
```

* Defining the splits.
The following commands define splits under cn=ibmpolicies, cn=aixdata and ou=people,cn=aixdata.

Listing 7. Defining different splits

```
========
ibmpolicies split definitions :
========

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=ibmpolicies split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
   cn: ibmpolicies split
ibm-slapdProxyNumPartitions: 1
ibm-slapdProxyPartitionBase: cn=ibmpolicies
ibm-slapdProxySplitName: ibmpoliciesplit
objectclass: top
objectclass: ibm-slapdConfigEntry
objectclass: ibm-slapdProxyBackendSplitContainer

# idsldapadd -h secldapproxy.ibm.com -D cn=root -w root
dn: cn=split1, cn=ibmpolicies split, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory, cn=Schemas, cn=Configuration
   cn: split1
ibm-slapdProxyBackendServerDN: cn=secldapinst1a,cn=ProxyDB,cn=Proxy Backends, cn=IBM Directory, cn=Schemas,cn=Configuration
```
**cn=IBM Directory,cn=Schemas,cn=Configuration**

ibm-slapdProxyBackendServerRole: any  
ibm-slapdProxyPartitionIndex: 1  
objectclass: top  
objectclass: ibm-slapdConfigEntry  
objectclass: ibm-slapdProxyBackendSplit

========

cn=aixdata split definitions :
========

# idsldapadd -h seclldaproxy.ibm.com -D cn=root -w root  
dn: cn=cn=aixdata, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,\  
cn=Schemas,\  
cn=Configuration  
cn: cn=aixdata  
ibm-slapdProxyNumPartitions: 1  
ibm-slapdProxyPartitionBase: cn=aixdata  
ibm-slapdProxyPartitionIndex: 1  
objectclass: top  
objectclass: ibm-slapdConfigEntry  
objectclass: ibm-slapdProxyBackendSplitContainer

# idsldapadd -h seclldaproxy.ibm.com -D cn=root -w root  
dn: cn=split1, cn=cn=aixdata, cn=ProxyDB, cn=Proxy Backends, cn=IBM Directory,\  
cn=Schemas, cn=Configuration  
cn: split1  
ibm-slapdProxyBackendServerDN: cn=secldapinst1a,cn=ProxyDB,cn=Proxy\  
Backends,\  
cn=IBM Directory,cn=Schemas,cn=Configuration  
ibm-slapdProxyBackendServerRole: any  
ibm-slapdProxyPartitionIndex: 1  
objectclass: top  
objectclass: ibm-slapdConfigEntry  
objectclass: ibm-slapdProxyBackendSplitContainer

========

cn=ou=people,cn=aixdata split
========

# idsldapadd -h seclldaproxy.ibm.com -D cn=root -w root  
dn: cn=ou=people,cn=aixdata, cn=ProxyDB, cn=Proxy Backends, cn=IBM\  
Directory,\  
cn=Schemas, cn=Configuration  
cn: ou=people,cn=aixdata  
ibm-slapdProxyNumPartitions: 2  
ibm-slapdProxyPartitionBase: ou=People,cn=aixdata  
ibm-slapdProxyPartitionIndex: 1  
objectclass: top  
objectclass: ibm-slapdConfigEntry  
objectclass: ibm-slapdProxyBackendSplitContainer

# idsldapadd -h seclldaproxy.ibm.com -D cn=root -w root  
dn: cn=split1, cn=ou=people,cn=aixdata, cn=ProxyDB, cn=Proxy\  
Backends,\  
cn=IBM Directory, cn=Schemas, cn=Configuration  
cn: split1  
ibm-slapdProxyBackendServerDN: cn=secldapinst1a,cn=ProxyDB,cn=Proxy\  
Backends,\  
cn=IBM Directory,cn=Schemas,cn=Configuration  
ibm-slapdProxyBackendServerRole: any  
ibm-slapdProxyPartitionIndex: 1  
objectclass: top  
objectclass: ibm-slapdConfigEntry  
objectclass: ibm-slapdProxyBackendSplit

# idsldapadd -h seclldaproxy.ibm.com -D cn=root -w root  
dn: cn=split2, cn=ou=people,cn=aixdata, cn=ProxyDB, cn=Proxy\  
Backends,
sectoldif - AIX local security to LDAP data migration

The sectoldif command is a migration tool provided with AIX for migrating local security data to LDAP. It can convert data from the local system directly to one of the three schemas: RFC2307, RFC2307AIX, or the AIX schema. The data is written to standard output. The LDIF file can then be imported to the LDAP server using the idsldapadd command.

**Listing 8. Converting system information to LDIF format**

```
# sectoldif -d cn=aixdata -S RFC2307 > /tmp/systeminfo.ldif
```

For loading data from the existing AIX server, run the sectoldif command so that the user/group and other information is transferred in the LDIF format. Once you have this information, you can directly load the output LDIF file to the proxy server.

- Use the idsldapadd command to load the data.
  Once you have all the data in the LDIF format, you can easily load that in your environment using the idsldapadd command. Execute this load operation on the proxy server. Adding the information on the proxy server will ensure that the any particular entry goes into the correct proxy split.

**Listing 9. Loading the data from LDIF**

```
# idsldapadd -h seclddaproxy.ibm.com -D cn=manager,cn=ibmpolicies -w sec001ret -c -i /tmp/systeminfo.ldif
```

- Use the mksecldap command to configure clients with the proxy.
  Now you are ready with the LDAP servers. You have a setup ready with the desired replication topology and also have a proxy server that the clients will contact.
Introducing ITDS proxy in an existing environment

In a typical production environment, the user repository is maintained in a centralized LDAP server. All clients authenticate using the LDAP server. High availability is maintained by having a peer-master LDAP server. The topology is shown in Figure 4.

Figure 4. Typical setup for authentication

With this setup in place, you need to get a ITDS proxy server in place in between the existing infrastructure and the clients. To start with, you will have to get the backup of the basic subtree (for example, cn=aixdata). Following are the steps:

1. Back up the data under ou=people,cn=aixdata. This subtree is going to have all of the users' data, like gidnumber, login shell, last login details,
and more.

**Listing 11. Get LDIF dump for ou=people,cn=aixdata**

```
# idsdb2ldif -I aixauth -s ou=people,cn=aixdata -o /tmp/people.ldif
```

2. Once you back up the data under ou=people,cn=aixdata, delete the same subtree, so that subsequent backups do not have copies.

**Listing 12. Delete data under ou=people,cn=aixdata from the server**

```
# idsldapdelete -D cn=root -w root -s ou=people,cn=aixdata
```

3. Back up the data under cn=aixdata. In this backup, there is no data under ou=people,cn=aixdata.

**Listing 13. Get LDIF dump for cn=aixdata**

```
# idsdb2ldif -I aixauth cn=aixdata -o /tmp/aixdata.ldif
```

4. Delete the data under cn=aixdata.

**Listing 14. Delete data under cn=aixdata from the server**

```
# idsldapdelete -D cn=root -w root -s cn=aixdata
```

5. Change the password encryption to crypt.

**Listing 15. Change default password encryption to crypt**

```
# idsldapadd -D cn=root -w root
dn: cn=Configuration
changetype: modify
replace: ibm-slapdPwEncryption
ibm-slapdPwEncryption: crypt
```

6. Create two new instances. You will use the existing setup of peer-peer replication to become one of the server group (say Server Group 1, as shown in Figure 1). The two new instances being added in this group will be used as another setup of servers, which will hold the complete cn=aixdata subtree, cn=ibmpolicies subtree and partial
Listing 16. Create two new instances on preferably two different servers

```
Add user aixauth
# idsadduser -g idsldap -l /home/aixauth -u aixauth -w aixauth

Create TDS RDBM instance
# idsicrt -I aixauth -e abcd1234567890 -l /home/aixauth

Configure database instance with this TDS instance
# idscfgdb -I aixauth -l /home/aixauth -a aixauth -w aixauth -t aixauth

Configure Admin username and password
# idsdnpw -I aixauth -u cn=root -p root

Configure cn=aixdata suffix
# idscfgsuf -I aixauth -s cn=aixdata

Start TDS instance to change default password encryption
# ibmslapd -I aixauth

Change default password encryption to crypt
# idsldapadd -D cn=root -w root
dn: cn=Configuration
changetype : modify
replace : ibm-slapdPwEncryption
ibm-slapdPwEncryption: crypt
```

7. The following steps must be taken to set up the environment. Since you have already learned the steps in detail in the previous section, we will just give a reference to the above section/code listing in this section when listing down the steps.

- Configure replication between the four TDS instances for cn=aixdata and cn=ibmpolicies. Two instances were from the existing environment and another two instances were created in Listing 14.

- Load data on any of the servers from /tmp/aixdata.ldif file. This LDIF file was collected from Listing 13.

- Add the missing ou=people,cn=aixdata entry only on the servers.

- Set up peer-to-peer replication topology as a group of two instances between secdapinst1a.ibm.com and secdapinst1b.ibm.com (as Server Group 1) and secdapinst2a.ibm.com and secdapinst2b.ibm.com (as Server Group 2) for the subtree ou=people,cn=aixdata. The topology should look like the one in Figure 3.

- Follow the steps mentioned in the section TDS Proxy configuration of
this article to configure and get a proxy server up for our usage.

• Add the data collected in aixdata.ldif from Listing 13.

8. **Listing 17. Loading the data from LDIF**

   idsldapadd command can be used as follows:

   ```shell
   # idsldapadd -h secldapproxy.ibm.com -D cn=manager,cn=ibmpolicies -w sec001ret 
   -c -i /tmp/aixdata.ldif
   ```

9. **mksecldap command to configure clients with the proxy.**

   **Listing 18. Command to configure a client**

   ```shell
   # mksecldap -c -h secldapproxy.ibm.com -a cn=manager,cn=ibmpolicies -p sec001ret
   ```
Resources

Learn

• **IBM Tivoli Directory Server documentation.** contains information on installing, configuring, administering, and using Tivoli Directory Server.

• Learn more about the **secldapclntd daemon**.

• **Introduction to ITDSv6.1 Proxy Server**

Get products and technologies

• Download **IBM product evaluation versions** and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

Discuss

• Participate in the AIX and UNIX forums:
  • **AIX Forum**
  • **AIX Forum for developers**
  • **Cluster Systems Management**
  • **IBM Support Assistant Forum**
  • **Performance Tools Forum**
  • **Virtualization Forum**
  • **More AIX and UNIX Forums**

• **developerWorks blogs** and get involved in the **developerWorks community**.

About the authors

Nikhil Firke
Nikhil is a System Software Engineer, currently working with Tivoli Directory Server Level 2 support team, IBM India Software Labs. He holds a degree in Computer Engineering from Pune Institute of Computer Technology, Pune (India).

Nilesh T. Patel
Nilesh is a System Software Engineer, currently working with Tivoli security team, IBM India Software Labs and he holds a degree in Information technology from Pune Institute of Computer Technology, Pune (India). His areas of expertise include IBM Tivoli Directory Server from the Tivoli Security Products and DB2. He is IBM Certified Deployment Professional for Tivoli Directory Server V6.1, Information Technology Infrastructure Library (ITIL) Foundation and DB2 UDB V8.1 certified.