LDAP Extensions for Proxy Connection
draft-vmodi-ldapext-proxy-connect-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.htm.

This Internet-Draft will expire on February 3, 2000.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

This document describes the LDAP extensions which allow LDAP client to connect to a DSA via an intermediate proxy / chain of proxies. (DSA is an X.500[1] term for directory server)

Specification Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119[3].

LDAP client is the software which provides access to LDAP directory via Lightweight Directory Access Protocol(LDAP) [2]. In this draft sometimes the word "client", is used for LDAP client.

DSA is the directory server which serves LDAP clients and allow them to access the directory information.
LDAP proxy server is the software which in itself is not the directory but can serve the LDAP client by connecting to DSA on behalf of client. In this draft sometimes the word "proxy" or "LDAP proxy" is used for LDAP proxy server.

Table of Contents

1. Introduction ................................................. 3
2. Problem Description ........................................ 4
3. Proposal Overview ........................................... 5
3.1 Usage ......................................................... 7
3.1.1 Client Configuration ................................. 7
3.1.2 Application Configuration .......................... 8
4. ASN.1 Definition of Extensions .......................... 9
4.1 ProxyConnectRequest ........................................ 9
4.1.1 Processing ProxyConnectRequest .................... 10
4.2 ProxyConnectResponse ....................................... 11
5. Security Consideration ..................................... 15
6. Acknowledgements ........................................... 16
   References ..................................................... 17
   Author’s Address ............................................. 17
   Full Copyright Statement .................................. 18
1. Introduction

Directory is at the heart of many powerful and distributed applications today. More and more information on Internet and Intranet is getting organized in directories. The protocol most widely used to access the directory information is LDAP[2]. As the number of LDAP client increases, the directory server (DSA) will be loaded more, hence the performance degradation. So there is a need to cache the directory information at the location other than the original directory server to improve the performance and provide disconnected operation. Also the organizations would like to monitor increasing LDAP traffic coming in to and/or going out of corporate Intranet for security and management reasons. So the LDAP client must be able to connect to DSA via intermediate software called proxy. This will ensure that all LDAP request and response exchanges between client and DSA happen via proxy, and thus allowing the proxy to monitor/control the LDAP traffic and/or cache the contents of LDAP requests and/or responses.
2. Problem Description

In LDAP, client opens an LDAP connection to DSA. Once the connection between LDAP client and DSA is established both of them uses that connection to exchange requests and responses. However, as discussed earlier it should be possible for LDAP client to be able to connect to DSA via one or more LDAP proxies. Version 3 of LDAP specification does not include specifications for the proxy connection.
3. Proposal Overview

The LDAP client instead of directly connecting to DSA, first connects to the LDAP proxy server and sends the ProxyConnectRequest, an extended LDAP request, the ASN.1 definition of which, is specified in this draft. The ProxyConnectRequest in addition to other parameters contains the URL of the DSA, using which the proxy tries to connect to it. The result of this operation is passed back to the client as ProxyConnectResponse. Once the connection between ‘client and proxy’ and ‘proxy and DSA’ is established, all the requests from client to DSA go via proxy and in the same manner all the responses from DSA to client go via proxy. Following Figure 1 shows the normal communication sequence between client and DSA.

```
+-------------+                     +-------------+
| LDAP Client |<------------------->|     DSA     |
+-------------+   LDAP Connection   +-------------+

Figure 1
Communication between LDAP Client and DSA
without LDAP Proxy Server
```
Following Figure 2 shows the communication between LDAP client and DSA via LDAP Proxy.

```
+-------------+      +------------+      +-------------+
| LDAP Client |<---->| LDAP Proxy |<---->|      DSA    |
+-------------+      +------------+      +-------------+
```

<table>
<thead>
<tr>
<th>Client</th>
<th>LDAP Proxy</th>
<th>DSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Open Connection -------&gt; Opens LDAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----- Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProxyConnect Request having URL of DSA -------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAP Open Connection -------&gt; Opens LDAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;----- Connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProxyConnect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------- Response</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAP Message (Request) -------&gt; LDAP Message (Requests) -------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAP Message (Response)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------ (Response)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;------ (Response)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>. . . -------&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>. . . -------&gt;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
This proposal defines an extended request, called "ProxyConnectRequest" and an extended response, called "ProxyConnectResponse" using which the LDAP client will be able to connect to DSA via LDAP proxy or chain of LDAP proxies.

3.1 Usage

There are two ways in which this proposal can be used. One requires changes in the existing client and the other one in the LDAP application.

3.1.1 Client Configuration

The LDAP client can be configured to make connection to LDAP proxy upon receiving LDAP open connection request (usually in the form of an API) from an application. It must then send ProxyConnectRequest having the URL of the DSA specified by the application in the LDAP open connection request, to the LDAP proxy. Depending upon the contents of the ProxyConnectResponse from proxy, client can decide the reply of LDAP open connection request to the application.

The client must have a way of specifying the proxy configuration parameters. The parameters are explained in ASN.1 definition of ProxyConnectRequest and ProxyConnectResponse.

The advantage of this configuration/usage is that the redirection of LDAP requests and responses is transparent to application. Applications need not know about proxy. However the disadvantage is...
that the existing clients need to be modified.

3.1.2 Application Configuration

To make use of proxy, application must ask the client to open an LDAP connection with proxy instead of DSA. The application then should send ProxyConnectRequest having URL of the DSA along with other parameters to proxy, before sending any other request. Depending upon the contents of ProxyConnectResponse application can decide to continue or close the connection with proxy. Depending upon the implementation of proxy, if the application, after opening the connection with proxy, sends any other request before sending ProxyConnectRequest and before getting valid response from proxy, the proxy may reply with an error, or may close the connection, or may serve the request from the cached entries of directory, if it has any.

The advantage of this configuration/usage is that the client need not change in order to make use of proxy. However the disadvantage is that the application must change and must know the details of proxy.
4. ASN.1 Definition of Extensions

This sections describes the ASN.1 definition of ProxyConnectRequest and ProxyConnectResponse. This release is the first attempt to capture all the required parameters of the extended request and response for proxy support. Comments on the design and utility of these definitions are specifically invited from all reviewers.

4.1 ProxyConnectRequest

The ASN.1 definition of ExtendedRequest as per LDAP(V3) specification[2] is as follows.

```
ExtendedRequest ::=  [APPLICATION 23] SEQUENCE {
  requestName  [0] LDAPOID,
  requestValue [1] OCTET STRING OPTIONAL
}
```

requestName(LDAPOID) of ProxyConnectRequest is TBD. The definition of requestValue is as follows.

```
requestValue ::=  SEQUENCE {
  dsaAddress              LDAPURL,
  requiredProxyRoute      SEQUENCE of LDAPURL,
  currentProxyRoute       SEQUENCE of LDAPURL,
  allowDynamicRouting     BOOLEAN DEFAULT FALSE
}
```

where,

```
LDAPURL ::= LDAPString -- limited to characters permitted in URLs
```

Parameters of ProxyConnectRequest’s requestValue are:

- dsaAddress: specifies the URL of the DSA(directory server) to which client wants to connect via the list of proxies specified in the requiredProxyRoute.

- requiredProxyRoute: specifies the URLs of the proxies via which to connect to DSA in the correct order. (This is usually known as chaining of proxies. Here, the client connects to one proxy which in turn connects to another proxy and so on, thus forming the chain of proxies. The chain ends with the connection to DSA.)

If the allowDynamicRouting flag is FALSE, this parameter must contain URL of the proxy receiving this request as the first entry in the sequence. However if the allowDynamicRouting flag is
- currentProxyRoute: contains the information about the current route from client to the proxy receiving this request. Each proxy after serving ProxyConnectRequest and before forwarding it to next proxy in the chain, adds its own URL to this parameter. This helps in determining the loopback in the proxy chain. This parameter must be set only by LDAP proxies and not by LDAP client or application.

- allowDynamicRouting: specifies whether the proxy can dynamically determine the next proxy in the chain during initial connection setup with DSA, for which this proposal is to be used. If the value of this parameter is TRUE, actual route between DSA and LDAP client obtained after ProxyConnectRequest and ProxyConnectResponse exchanges can be different than the one specified in the requiredProxyRoute parameter.

LDAP client/application specifies the list of proxies via which to connect to DSA in requiredProxyRoute parameter. This require client/application to know all the proxies in the chain. However this is not usually the case because ..

1. client/application may know only the immediate proxy.

2. proxy chain or route to different DSAs will most likely to be different.

In such cases allowDynamicRouting flag should be set to TRUE. This gives flexibility to proxies in determining the next proxy in the chain.

4.1.1 Processing ProxyConnectRequest

Each LDAP proxy server upon receiving ProxyConnectRequest must ..

1. verify that servicing of the received ProxyConnectRequest is not going to form loopback in the proxy chain. This can be done by checking the presence of its URL in currentProxyRoute. If there is a possibility of loopback the proxy must terminate the connection from which the ProxyConnectRequest is received; either, without sending the error message or after sending the error message via ProxyConnectResponse.

2. remove its URL from requiredProxyRoute list, if present. Add its URL to the currentProxyRoute list.

3. determine the next proxy in the chain.
If allowDynamicRouting flag is FALSE, and requiredProxyRoute list is not empty, the next proxy in the chain is the first entry in the list.
If allowDynamicRouting flag is FALSE, and requiredProxyRoute list is empty the proxy chain ends.
If allowDynamicRouting flag is TRUE, the proxy can determine the next proxy in the chain from its configuration parameters or by some other mechanism. OR proxy may decide to end the proxy chain.

4. try to connect to the next proxy if identified, and send the ProxyConnectRequest to it, if connected and wait for ProxyConnectResponse from it. From the status of the connection and/or contents of the ProxyConnectResponse from the next proxy, create ProxyConnectResponse and send it to the previous proxy/LDAP client connected to it. OR connect to DSA if next proxy is not identified. From the status of the connection create ProxyConnectResponse and send it to the previous proxy/LDAP client connected to it.

4.2 ProxyConnectResponse

LDAP proxy must reply to ProxyConnectRequest received from LDAP client or previous proxy by sending ProxyConnectResponse. The response must include the result of opening the LDAP connection with next proxy or DSA. It must also contain the necessary information about the proxy server. Components of ProxyConnectResponse obtained from the later proxies in the chain are also passed on to the previous proxies to make sure that client receives information about all the proxies and connections via which it is connected or attempted to connect to DSA.

The ASN.1 definition of ExtendedResponse as per LDAP(V3) specification[2] is as follows.

ExtendedResponse ::=  [APPLICATION 24] SEQUENCE {
  COMPONENTS OF LDAPResult,
  responseName [10] LDAPOID OPTIONAL,
}

responseName(LDAPOID) of ProxyConnectResponse is TBD. The definition of response is as follows.
response ::= SEQUENCE {
    proxyServerInfo   ProxyServerInformation,
    connection        ProxyConnection,
    previousResponses PreviousProxyConnectResponses
}

where,
- proxyServerInfo: gives various information about proxy server like its URL, brand information, cache parameters - if available etc. Following is the ASN.1 definition of ProxyServerInformation.

ProxyServerInformation ::= SEQUENCE {
    proxyAddress      LDAPURL,
    cacheAvailability BOOLEAN,
    brandInfo         BrandInformation,
    cacheAttrs        CacheAttributes OPTIONAL
}

where,
- proxyAddress: is the URL of the proxy server.
- cacheAvailability specifies whether the proxy can cache contents of LDAP requests and responses or not.
- brandInfo: identifies the name of the proxy server software, its version, name of the company owning it and contact information such as email, phone number, URL etc.

BrandInformation ::= SEQUENCE {
    name         LDAPString,
    version      LDAPString,
    organization LDAPString,
    contactInfo  LDAPString
}

- cacheAttrs: this parameter must be present if cacheAvailability flag is TRUE. It must be ignored if cacheAvailability flag is FALSE. This parameter specifies various cache parameters. The ASN.1 definition of CacheAttributes is as follows.
CacheAttributes ::= SEQUENCE {
  identification OCTET STRING,
    /* Cache Identification */
  type ENUMERATED {
    readOnly (0), /* Cache is readOnly */
    readWrite (1) /* Cache is both read and write */
  },
  storage ENUMERATED {
    transient (0), /* Cached objects are transient */
    persistent (1) /* Cached objects are persistent */
  },
  encryptedStore BOOLEAN DEFAULT FALSE,
    /* Cached object storage is encrypted. */
  authorizationSupport BOOLEAN DEFAULT FALSE
    /* cache supports authorization and access control */
}

- connection: describes the status of the LDAP connection from the proxy whose information is specified in proxyServerInfo to either next proxy in the chain or DSA. Following is the ASN.1 definition of ProxyConnection.

ProxyConnection ::= SEQUENCE {
  dsaAddress LDAPURL,
  connectionState ENUMERATED {
    notConnected (0), /* Could not connect to the * ldapServer. */
    connected (1), /* Connected to ldapServer. */
    lazyConnect (2), /* will connect on * receiving any further * request. */
  }
}

- previousResponses: Each proxy in the chain sends ProxyConnectRequest to next proxy in the chain. Necessary components of the ProxyConnectResponse from the next proxy must be included in the ProxyConnectResponse to the previous proxy/client connected to it. Thus this parameter contains components of ProxyConnectResponse from each proxy ahead of this proxy in the chain. Following is the ASN.1 definition of PreviousProxyConnectResponses.
PreviousProxyConnectResponses ::=  
  SEQUENCE OF PreviousProxyConnectResponse

PreviousProxyConnectResponse
  {
    COMPONENTS OF LDAPResult COMPONENT OF ProxyConnectResponse,  
    proxyServerInfo COMPONENT OF ProxyConnectResponse,  
    connection COMPONENT OF ProxyConnectResponse
  }

ProxyConnectResponse generated by last proxy in the chain which is connected to DSA will have no entries in PreviousProxyConnectResponses parameter. Every other proxy in the chain upon receiving ProxyConnectResponse from the proxy ahead in the chain, must copy the entries of PreviousProxyConnectResponses from it to the same parameter of ProxyConnectResponse it is creating. Also it must add the ProxyConnectResponse entry containing the required components of ProxyConnectResponse received from proxy ahead in the chain to the PreviousProxyConnectResponses parameter of the ProxyConnectResponse it is creating.
5. Security Consideration

When LDAP client is connected to DSA via one or more proxies, each request (including BindRequest) and response goes via the proxies. Thus the proxy has access to all the information within requests and responses. The client must trust the proxy that it will not misuse the information knowingly or otherwise. The client should verify the ProxyConnectResponse and should terminate the connection in case it does not trust any proxy in the chain. Also if client intends to access sensitive information from the DSA and if there are proxies in the chain which caches the information but does not provide authorization and access control to the cached entries, it should terminate the connection.

However the proxy can provide false information to the client in ProxyConnectResponse, so the ultimate security is the trust relationship.
6. Acknowledgements

I sincerely thank Ed Reed for reviewing this proposal and providing valuable suggestions. I also thank my team members Anasuya Devi, Ganesh S P, Deepa P and Vijay K N, for reviewing the early versions of this document and providing valuable comments. Special thanks to Ganesh S P and Deepa P for pointing out grammatical and spelling errors.
Internet-Draft    LDAP Extensions for Proxy Connection       August 1999

References


Author’s Address

Vipul Modi
Novell, Inc.
Novell Product Group Bangalore
49/1 & 49/3 G. B. Pallya
7th Mile, Hosur Road
Bangalore, Karnataka  560068
India

Phone: +91 80 572 1876/62/56 Ext:2015
Fax: +91 80 572 180
EMail: mvipul@novell.com
URI:  http://www.novell.com
Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to
others, and derivative works that comment on or otherwise explain it
or assist in its implementation may be prepared, copied, published
and distributed, in whole or in part, without restriction of any
kind, provided that the above copyright notice and this paragraph
are included on all such copies and derivative works. However, this
document itself may not be modified in any way, such as by removing
the copyright notice or references to the Internet Society or other
Internet organizations, except as needed for the purpose of
developing Internet standards in which case the procedures for
copyrights defined in the Internet Standards process must be
followed, or as required to translate it into languages other than
English.

The limited permissions granted above are perpetual and will not be
revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an
"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC editor function is currently provided by the
Internet Society.