Network Working Group
Internet-Draft
Expires: November 8, 2001

Johansson
Stockholm University
Hedberg
Catalogix
May 10, 2001

Lightweight Directory Access Protocol over UDP/IP
draft-ietf-ldapext-ldapudp-00

Status of this Memo

Copyright Notice

Abstract

This memo describes modifications to LDAP version 3[1] to allow transport of a subset of the LDAP protocol over UDP/IP.

Table of Contents

1. Overview and Rationale

Using LDAP version 3[1] involves normal TCP/IP connection setup which for some applications may constitute undesirable overhead, especially in situations where only unauthenticated requests are performed. The typical use would be for fast light-weight read-only clients where the number of round-trips must be kept to a minimum or for clients which makes large numbers of requests to multiple LDAP servers. An example of the latter would be an LDAP server which maintains a CIP[6] index and provides chaining of requests to servers indexed by the mesh. Such a server will often have to maintain large numbers of tcp connections. Experience from the TISDAG[5] project has shown that even with relatively small indices and few concurrent clients to the index server the number of outgoing tcp connections may be very large.

2. Protocol Elements and Result Codes

The protocol messages of LDAPv3/UDP are identical with those of LDAPv3 and each LDAPMessage is encoded and transmitted in a single UDP datagram. In addition a new result code is defined:

connectionRequired	(70??)

The semantics of this result code is as follows:

* Whenever a server implementing the protocol described in this draft or any protocol derived from this protocol receives a request it for some reason is unwilling or unable to perform over connection-less transport the server must return this result code. Typical examples for this are when the resultset is to large to fit into the biggest packet the network in use can support or when a client tries to do a bind but does not provide enough information for it to succeed

* Whenever a client implementing the protocol described in this draft or any protocol derived from this protocol receives this result code the client must not retry the request using connection-less transport.

3. Description of the protocol

Use of the LDAPv3 protocol over UDP means that protocol elements can become dropped, delayed or even duplicated by the transport layer. In order to deal with these situations clients and servers implementing this protocol must employ some means for detecting and/or retrying failed requests.

Note that the search operation is slightly different in this respect. A SearchResultEntry or a SearchResultReference can become lost or duplicated withouth affecting the flow of requests and responses between the client and the server as long as the SearchResultDone packet is not lost. The loss of this packet would be indistinguishable from the situation where the search is still underway. Thus the delivery of the resultcode packets (including the ExtendedResponse) is different from the delivery of search result packets. Since the application may or may not care about actually receiving SearchResultEntry and SearchResultReference packets some method for ensuring the delivery of these may or may not be needed.

The reason why the safe delivery of the result-code pdu is important can be illustrated with a simple example. Assume that a client issues an add operation for a new entry. This request is received by the server and the add operation is performed but the resultcode (SUCCESS) gets lost on its way to the client. If the client were to retry the operation by issuing the same add request under a new message id the result code would indicate a failure since the object already exists in the server.

There are several possible mechanisms for solving the problems described above and a particular choice must be agreed upon by the client and server before using ldap over connection-less transports. The method by which a mechanism is selected is not covered by this document but may involve the client connecting to the server over tcp to read the root-DSE entry before using connection-less transport. This standard may be extended by specifying other mechanisms for safe delivery of protocol messages.

Servers implementing this protocol SHOULD provide a protocol listener on port 389. How the existence of other protocol listeners are communicated to clients (server location) is not covered in this document.

To be used over LDAPv3/UDP other extensions defined for LDAPv3 must be amended by text which explains how the controls and/or exops defined in the extension interact with LDAPv3/UDP. In particular, for each control that is marked critical by the extension the standard must explain how safe delivery of the pdu containing the

control is ensured.

4. Dealing with lost result PDUs: reuse of messageIDs

A simple method for sending and receiving protocol messages over lossy connection-less transport is reuse of messageIDs. Whenever a client times out before receiving a result PDU it is waiting for it may, using this mechanism, retry the same request using the same messageID as before. A server implementing reuse of messageIDs is required to maintain a cache (the size of which should be annonced in the rootDSE-object; see below) of recent result-codes for each source port and address. Consequently a client using this mechanism must bind to the local port before issuing requests so that a particular client process can be identified by the server. The client must not issue more operations at a time than the cachesize.

A server implementing this mechanism must announce it by providing a value for the size of the result code cache in the root-DSE attribute LDAPResultCacheSize:

```
(<TBA>
 NAME 'LDAPResultCacheSize'
 DESC 'The size of the per-client cache of resultcodes
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
 EQUALITY 'integerMatch'
 NO-USER-MODIFICATION
 USAGE dSAOperation )
```

Note that this mechanism does not protect agains a third party inserting protocol messages. See the section on security considerations.

5. Security considerations

Since SASL[3] is only defined for connection-oriented operation it is not possible to use SASL authentication with LDAPv3/UDP and a server must respond with an result code of connectionRequired (??) if a bind requesting SASL authentication is received.

Mechanisms for safe delivery of protocol messages which do not protect against third-party attacks (inserting messages into the protocol stream) should not be used for update operations unless the underlying transport provides protection against such attacks.

References

[1] Wahl, M., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[2] Kent, S and R Atkinson, "Security Architecture for the Internet Protocol", November 1998.

[3] Myers, J, "Simple Authentication and Security Layer (SASL)", October 1997.

[4] Armijo, M. P., Esibov, L. and P. Leach, "Discovering LDAP Services with DNS", Internet-Draft draft-ietf-ldapext-locate-02, April 2000.

[5] Hedberg, R and L Daigle, "Technical Infrastructure for Swedish Directory Access Gateways (TISDAG)", January 2000.

[6] Hedberg, R, "LDAPv2 client vs. the Index Mesh", RFC 2657, August 1999.

Authors' Addresses

Leif Johasson
Stockholm University
Stockholm SE-10691
Sweden

Phone: +46 8 164541
EMail: leifj@it.su.se

Roland Hedberg
Catalogix
Jegerveien 25
Oslo 0777
Norway

Phone: +47 23082996
EMail: roland@catalogix.se

Full Copyright Statement

Acknowledgement

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