Network Working Group
Request for Comments: 2255
Category: Standards Track

T. Howes
M. Smith
Netscape Communications Corp.

December 1997

The LDAP URL Format

1. Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

IESG NOTE

This document describes a directory access protocol that provides both read and update access. Update access requires secure authentication, but this document does not mandate implementation of any satisfactory authentication mechanisms.

In accordance with RFC 2026, section 4.4.1, this specification is being approved by IESG as a Proposed Standard despite this limitation, for the following reasons:

a. to encourage implementation and interoperability testing of these protocols (with or without update access) before they are deployed, and

b. to encourage deployment and use of these protocols in read-only applications. (e.g. applications where LDAPv3 is used as a query language for directories which are updated by some secure mechanism other than LDAP), and

c. to avoid delaying the advancement and deployment of other Internet standards-track protocols which require the ability to query, but not update, LDAPv3 directory servers.

Readers are hereby warned that until mandatory authentication mechanisms are standardized, clients and servers written according to this specification which make use of update functionality are UNLIKELY TO INTEROPERATE, or MAY INTEROPERATE ONLY IF AUTHENTICATION IS REDUCED TO AN UNACCEPTABLY WEAK LEVEL.

Implementors are hereby discouraged from deploying LDAPv3 clients or servers which implement the update functionality, until a Proposed Standard for mandatory authentication in LDAPv3 has been approved and published as an RFC.

2. Abstract

LDAP is the Lightweight Directory Access Protocol, defined in [1], [2] and [3]. This document describes a format for an LDAP Uniform Resource Locator. The format describes an LDAP search operation to perform to retrieve information from an LDAP directory. This document replaces RFC 1959. It updates the LDAP URL format for version 3 of LDAP and clarifies how LDAP URLs are resolved. This document also defines an extension mechanism for LDAP URLs, so that future documents can extend their functionality, for example, to provide access to new LDAPv3 extensions as they are defined.

The key words "MUST", "MAY", and "SHOULD" used in this document are to be interpreted as described in [6].

3. URL Definition

An LDAP URL begins with the protocol prefix "ldap" and is defined by the following grammar.

```
ldapurl    = scheme "://" [hostport] ["/"
             [dn ["?" [attributes] ["?" [scope]
             ["?" [filter] ["?" extensions]]]]]]
scheme     = "ldap"
attributes = attrdesc *("," attrdesc)
scope      = "base" / "one" / "sub"
dn         = distinguishedName from Section 3 of [1]
hostport   = hostport from Section 5 of RFC 1738 [5]
attrdesc   = AttributeDescription from Section 4.1.5 of [2]
filter     = filter from Section 4 of [4]
extensions = extension *("," extension)
extension  = ["!"] extype ["=" exvalue]
extype     = token / xtoken
exvalue    = LDAPString from section 4.1.2 of [2]
token      = oid from section 4.1 of [3]
xtoken     = ("X-" / "x-") token
```

The "ldap" prefix indicates an entry or entries residing in the LDAP server running on the given hostname at the given portnumber. The default LDAP port is TCP port 389. If no hostport is given, the client must have some apriori knowledge of an appropriate LDAP server to contact.

The dn is an LDAP Distinguished Name using the string format described in [1]. It identifies the base object of the LDAP search.


```
ldapurl    = scheme "://" [hostport] ["/"
                 [dn ["?" [attributes] ["?" [scope]
                 ["?" [filter] ["?" extensions]]]]]]
    scheme     = "ldap"
    attributes = attrdesc *("," attrdesc)
    scope      = "base" / "one" / "sub"
    dn         = distinguishedName from Section 3 of [1]
    hostport   = hostport from Section 5 of RFC 1738 [5]
    attrdesc   = AttributeDescription from Section 4.1.5 of [2]
    filter     = filter from Section 4 of [4]
    extensions = extension *("," extension)
    extension  = ["!"] extype ["=" exvalue]
    extype     = token / xtoken
    exvalue    = LDAPString from section 4.1.2 of [2]
    token      = oid from section 4.1 of [3]
    xtoken     = ("X-" / "x-") token
```

The "ldap" prefix indicates an entry or entries residing in the LDAP server running on the given hostname at the given portnumber. The default LDAP port is TCP port 389. If no hostport is given, the client must have some apriori knowledge of an appropriate LDAP server to contact.

The dn is an LDAP Distinguished Name using the string format described in [1]. It identifies the base object of the LDAP search.

The attributes construct is used to indicate which attributes should be returned from the entry or entries. Individual attrdesc names are as defined for AttributeDescription in [2]. If the attributes part is omitted, all user attributes of the entry or entries should be requested (e.g., by setting the attributes field AttributeDescriptionList in the LDAP search request to a NULL list, or (in LDAPv3) by requesting the special attribute name "*").

The scope construct is used to specify the scope of the search to perform in the given LDAP server. The allowable scopes are "base" for a base object search, "one" for a one-level search, or "sub" for a subtree search. If scope is omitted, a scope of "base" is assumed.

The filter is used to specify the search filter to apply to entries within the specified scope during the search. It has the format specified in [4]. If filter is omitted, a filter of "(objectClass=*)" is assumed.

The extensions construct provides the LDAP URL with an extensibility mechanism, allowing the capabilities of the URL to be extended in the future. Extensions are a simple comma-separated list of type=value pairs, where the =value portion MAY be omitted for options not requiring it. Each type=value pair is a separate extension. These LDAP URL extensions are not necessarily related to any of the LDAPv3 extension mechanisms. Extensions may be supported or unsupported by the client resolving the URL. An extension prefixed with a '!' character (ASCII 33) is critical. An extension not prefixed with a ' !' character is non-critical.

If an extension is supported by the client, the client MUST obey the extension if the extension is critical. The client SHOULD obey supported extensions that are non-critical.

If an extension is unsupported by the client, the client MUST NOT process the URL if the extension is critical. If an unsupported extension is non-critical, the client MUST ignore the extension.

If a critical extension cannot be processed successfully by the client, the client MUST NOT process the URL. If a non-critical extension cannot be processed successfully by the client, the client SHOULD ignore the extension.

Extension types prefixed by "X-" or "x-" are reserved for use in bilateral agreements between communicating parties. Other extension types MUST be defined in this document, or in other standards-track documents.

One LDAP URL extension is defined in this document in the next section. Other documents or a future version of this document MAY define other extensions.

Note that any URL-illegal characters (e.g., spaces), URL special characters (as defined in section 2.2 of RFC 1738) and the reserved character '?' (ASCII 63) occurring inside a dn, filter, or other element of an LDAP URL MUST be escaped using the % method described in RFC 1738 [5]. If a comma character ',' occurs inside an extension value, the character MUST also be escaped using the % method.

4. The Bindname Extension

This section defines an LDAP URL extension for representing the distinguished name for a client to use when authenticating to an LDAP directory during resolution of an LDAP URL. Clients MAY implement this extension.

The extension type is "bindname". The extension value is the distinguished name of the directory entry to authenticate as, in the same form as described for dn in the grammar above. The dn may be the NULL string to specify unauthenticated access. The extension may be either critical (prefixed with a '!' character) or non-critical (not prefixed with a '!' character).

If the bindname extension is critical, the client resolving the URL MUST authenticate to the directory using the given distinguished name and an appropriate authentication method. Note that for a NULL distinguished name, no bind MAY be required to obtain anonymous access to the directory. If the extension is non-critical, the client MAY bind to the directory using the given distinguished name.

5. URL Processing

This section describes how an LDAP URL SHOULD be resolved by a client.

First, the client obtains a connection to the LDAP server referenced in the URL, or an LDAP server of the client's choice if no LDAP server is explicitly referenced. This connection MAY be opened specifically for the purpose of resolving the URL or the client MAY reuse an already open connection. The connection MAY provide confidentiality, integrity, or other services, e.g., using TLS. Use of security services is at the client's discretion if not specified in the URL.

Next, the client authenticates itself to the LDAP server. This step is optional, unless the URL contains a critical bindname extension with a non-NULL value. If a bindname extension is given, the client proceeds according to the section above.

If a bindname extension is not specified, the client MAY bind to the directory using a appropriate dn and authentication method of its own choosing (including NULL authentication).

Next, the client performs the LDAP search operation specified in the URL. Additional fields in the LDAP protocol search request, such as sizelimit, timelimit, deref, and anything else not specified or defaulted in the URL specification, MAY be set at the client's discretion.

Once the search has completed, the client MAY close the connection to the LDAP server, or the client MAY keep the connection open for future use.

6. Examples

The following are some example LDAP URLs using the format defined above. The first example is an LDAP URL referring to the University of Michigan entry, available from an LDAP server of the client's choosing:

```
ldap:///o=University%20of%20Michigan,c=US
```

The next example is an LDAP URL referring to the University of Michigan entry in a particular ldap server:

```
ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US
```

Both of these URLs correspond to a base object search of the "o=University of Michigan, c=US" entry using a filter of "(objectclass=*)", requesting all attributes.

The next example is an LDAP URL referring to only the postalAddress attribute of the University of Michigan entry:

```
ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,
       c=US?postalAddress
```

The corresponding LDAP search operation is the same as in the previous example, except that only the postalAddress attribute is requested.

The next example is an LDAP URL referring to the set of entries found by querying the given LDAP server on port 6666 and doing a subtree search of the University of Michigan for any entry with a common name of "Babs Jensen", retrieving all attributes:

```
ldap://host.com:6666/o=University%20of%20Michigan,
       c=US??sub?(cn=Babs%20Jensen)
```

The next example is an LDAP URL referring to all children of the c=GB entry:

```
ldap://ldap.itd.umich.edu/c=GB?objectClass?one
```

The objectClass attribute is requested to be returned along with the entries, and the default filter of "(objectclass=*)" is used.

The next example is an LDAP URL to retrieve the mail attribute for the LDAP entry named "o=Question?,c=US" is given below, illustrating the use of the escaping mechanism on the reserved character '?'.

```
ldap://ldap.question.com/o=Question%3f,c=US?mail
```

The next example illustrates the interaction between LDAP and URL quoting mechanisms.

```
ldap://ldap.netscape.com/o=Babsco,c=US??(int=%5c00%5c00%5c00%5c04)
```

The filter in this example uses the LDAP escaping mechanism of \ to encode three zero or null bytes in the value. In LDAP, the filter would be written as (int=\00\00\00\04). Because the \ character must be escaped in a URL, the \'s are escaped as %5c in the URL encoding.

The final example shows the use of the bindname extension to specify the dn a client should use for authentication when resolving the URL.

```
ldap:///??sub??bindname=cn=Manager%2co=Foo
ldap:///??sub??!bindname=cn=Manager%2co=Foo
```

The two URLs are the same, except that the second one marks the bindname extension as critical. Notice the use of the % encoding method to encode the comma in the distinguished name value in the bindname extension.

7. Security Considerations

General URL security considerations discussed in [5] are relevant for LDAP URLs.

The use of security mechanisms when processing LDAP URLs requires particular care, since clients may encounter many different servers via URLs, and since URLs are likely to be processed automatically, without user intervention. A client SHOULD have a user-configurable policy about which servers to connect to using which security mechanisms, and SHOULD NOT make connections that are inconsistent with this policy.

Sending authentication information, no matter the mechanism, may violate a user's privacy requirements. In the absence of specific policy permitting authentication information to be sent to a server, a client should use an anonymous connection. (Note that clients conforming to previous LDAP URL specifications, where all connections are anonymous and unprotected, are consistent with this specification; they simply have the default security policy.)

Some authentication methods, in particular reusable passwords sent to the server, may reveal easily-abused information to the remote server or to eavesdroppers in transit, and should not be used in URL processing unless explicitly permitted by policy. Confirmation by the human user of the use of authentication information is appropriate in many circumstances. Use of strong authentication methods that do not reveal sensitive information is much preferred.

The LDAP URL format allows the specification of an arbitrary LDAP search operation to be performed when evaluating the LDAP URL. Following an LDAP URL may cause unexpected results, for example, the retrieval of large amounts of data, the initiation of a long-lived search, etc. The security implications of resolving an LDAP URL are the same as those of resolving an LDAP search query.

8. Acknowledgements

The LDAP URL format was originally defined at the University of Michigan. This material is based upon work supported by the National Science Foundation under Grant No. NCR-9416667. The support of both the University of Michigan and the National Science Foundation is gratefully acknowledged.

Several people have made valuable comments on this document. In particular RL "Bob" Morgan and Mark Wahl deserve special thanks for their contributions.

9. References

[1] Wahl, M., Kille, S., and T. Howes, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names", RFC 2253, December 1997.

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

[3] Wahl, M., Coulbeck, A., Howes, T. and S. Kille, "Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

[4] Howes, T., "A String Representation of LDAP Search Filters", RFC 2254, December 1997.

[5] Berners-Lee, T., Masinter, L. and M. McCahill, "Uniform Resource Locators (URL)," RFC 1738, December 1994.

[6] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels," RFC 2119, March 1997.

Authors' Addresses

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA

Phone: +1 415 937-3419
EMail: howes@netscape.com

Mark Smith
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA

Phone: +1 415 937-3477
EMail: mcs@netscape.com

Full Copyright Statement