Internet X.509 Public Key Infrastructure
Operational Protocols - LDAPv2
<draft-ietf-pkix-opp-LDAPv2-08.txt>

1. Status of this Memo

This document is an Internet-Draft. 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 docu-
ments at any time. It is inappropriate to use Internet-Drafts as
reference material or to cite them other than as "work in pro-
gress."

To learn the current status of any Internet-Draft, please check the
"1id-abstracts.txt" listing contained in the Internet-Drafts Shadow
Directories on ftp.is.co.za (Africa), ftp.nordu.net   (Europe),
munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or
ftp.isi.edu (US West Coast).

2. Abstract

The protocol described in this document is designed to satisfy some
of the operational requirements within the Internet X.509 Public
Key Infrastructure (IPKI). Specifically, this document addresses
requirements to provide access to Public Key Infrastructure (PKI)
repositories for the purposes of retrieving PKI information and
managing that same information. The mechanism described in this
document is based on the Lightweight Directory Access Protocol
(LDAP) v2, defined in RFC 1777, defining a profile of that protocol
for use within the IPKI and updates encodings for certificates and
revocation lists from RFC 1778. Additional mechanisms addressing
PKIX operational requirements are specified in separate documents.

The key words 'MUST', 'REQUIRED', 'SHOULD', 'RECOMMENDED', and
'MAY' in this document are to be interpreted as described in RFC
2119.

Please send comments on this document to the ietf-pkix@imc.org mail
list.

3. Introduction

This specification is part of a multi-part standard for development
of a Public Key Infrastructure (PKI) for the Internet. This specif-
ication addresses requirements to provide retrieval of X.509 PKI
information, including certificates and CRLs from a repository.
This specification also addresses requirements to add, delete and
modify PKI information in a repository. A profile based on the LDAP
version 2 protocol is provided to satisfy these requirements.

4. Model

The PKI components, as defined in PKIX Part 1, which are involved
in PKIX operational protocol interactions include:

   - End Entities
   - Certification Authorities (CA)
   - Repository

End entities and CAs using LDAPv2, retrieve PKI information from
the repository using a subset of the LDAPv2 protocol.

CAs populate the repository with PKI information using a subset of
the LDAPv2 protocol.

5. Lightweight Directory Access Protocol (LDAP)

The following sections examine the retrieval of PKI information
from a repository and management of PKI information in a reposi-
tory. A profile of the LDAPv2 protocol is defined for providing
these services.

Section 6 satisfies the requirement to retrieve PKI information (a
certificate, CRL, or other information of interest) from an entry
in the repository, where the retrieving entity (either an end
entity or a CA) has knowledge of the name of the entry. This is
termed "repository read".

Section 7 satisfies the same requirement as 6 for the situation
where the name of the entry is not known, but some other related
information which may optionally be used as a filter against candi-
date entries in the repository, is known. This is termed "reposi-
tory search".

Section 8 satisfies the requirement of CAs to add, delete and
modify PKI information information (a certificate, CRL, or other
information of interest)in the repository. This is termed "reposi-
tory modify".

The subset of LDAPv2 needed to support each of these functions is
described below. Note that the repository search service is a
superset of the repository read service in terms of the LDAPv2
functionality needed.

Note that all tags are implicit by default in the ASN.1 defini-
tions that follow.

6. LDAP Repository Read

To retrieve information from an entry corresponding to the subject
or issuer name of a certificate, requires a subset of the following
three LDAP operations:

```
BindRequest (and BindResponse)
SearchRequest (and SearchResponse)
UnbindRequest
```

The subset of each REQUIRED operation is given below.

6.1. Bind

6.1.1. Bind Request

The full LDAP v2 Bind Request is defined in RFC 1777.

An application providing a LDAP repository read service MUST imple-
ment the following subset of this operation:

```
BindRequest ::=
  [APPLICATION 0] SEQUENCE {
     version      INTEGER (2),
     name         LDAPDN, -- MUST accept NULL LDAPDN
     simpleauth [0] OCTET STRING  -- MUST accept NULL simple
     }
```

An application providing a LDAP repository read service MAY imple-
ment other aspects of the BindRequest as well.

Different services may have different security requirements. Some
services may allow anonymous search, others may require authentica-
tion. Those services allowing anonymous search may choose only to
allow search based on certain criteria and not others.

A LDAP repository read service SHOULD implement some level of
anonymous search access. A LDAP repository read service MAY imple-
ment authenticated search access.

6.1.2. Bind Response

The full LDAPv2 BindResponse is described in RFC 1777.

An application providing a LDAP repository read service MUST imple-
ment this entire protocol element, though only the following error
codes may be returned from a Bind operation:

```
   success                    (0),
   operationsError            (1),
   protocolError              (2),
   authMethodNotSupported     (7),
   noSuchObject               (32),
   invalidDNSyntax            (34),
   inappropriateAuthentication  (48),
   invalidCredentials         (49),
   busy                       (51),
   unavailable                (52),
   unwillingToPerform         (53),
   other                      (80)
```

6.2. Search

6.2.1. Search Request

The full LDAPv2 SearchRequest is defined in RFC 1777.

An application providing a LDAP repository read service MUST imple-
ment the following subset of the SearchRequest.

```
   SearchRequest ::=
     [APPLICATION 3] SEQUENCE {
        baseObject      LDAPDN,
        scope              ENUMERATED {
                           baseObject   (0),
                                     },
        derefAliases    ENUMERATED {
                           neverDerefAliases   (0),
                                     },
        sizeLimit       INTEGER (0),
        timeLimit       INTEGER (0),
        attrsOnly       BOOLEAN, -- FALSE only
        filter          Filter,
        attributes      SEQUENCE OF AttributeType
                                }

   Filter ::=
     CHOICE {
```

```
             present          [7] AttributeType, -- "objectclass" only
                       }
```

This subset of the LDAPv2 SearchRequest allows the LDAPv2 "read"
operation: a base object search with a filter testing for the
existence of the objectClass attribute.

An application providing a LDAP repository read service MAY imple-
ment other aspects of the SearchRequest as well.

6.2.2.

The full LDAPv2 SearchResponse is defined in RFC 1777.

An application providing a LDAP repository read service over LDAPv2
MUST implement the full SearchResponse.

Note that in the case of multivalued attributes such as userCerti-
ficate a SearchResponse containing this attribute will include all
values, assuming the requester has sufficient access permissions.
The application/relying party may need to select an appropriate
value to be used. Also note that retrieval of a certificate from a
named entry does not guarantee that the certificate will include
that same Distinguished Name (DN) and in some cases the subject DN
in the certificate may be NULL.

6.3. Unbind

The full LDAPv2 UnbindRequest is defined in RFC 1777.

An application providing a LDAP repository read service MUST imple-
ment the full UnbindRequest.

7. LDAP Repository Search

To search ,using arbitrary criteria, for an entry in a repository
containing a certificate, CRL, or other information of interest,
requires a subset of the following three LDAP operations:

```
  BindRequest (and BindResponse)
  SearchRequest (and SearchResponse)
  UnbindRequest
```

The subset of each operation REQUIRED is given below.

7.1. Bind

The BindRequest and BindResponse subsets needed are the same as

those described in Section 6.1.

The full LDAP v2 Bind Request is defined in RFC 1777.

7.2. Search

7.2.1. Search Request

The full LDAPv2 SearchRequest is defined in RFC 1777.

An application providing a LDAP repository search service MUST
implement the following subset of the SearchRequest protocol unit.

```
SearchRequest ::=
  [APPLICATION 3] SEQUENCE {
     baseObject     LDAPDN,
     scope          ENUMERATED {
                          baseObject     (0),
                          singleLevel    (1),
                          wholeSubtree   (2)
                                },
     derefAliases   ENUMERATED {
                          neverDerefAliases    (0),
                                },
     sizeLimit      INTEGER (0 .. maxInt),
     timeLimit      INTEGER (0 .. maxInt),
     attrsOnly      BOOLEAN,  -- FALSE only
     filter         Filter,
     attributes     SEQUENCE OF AttributeType
                            }
```

All aspects of the SearchRequest MUST be supported, except for the
following:

- Only the neverDerefAliases value of derefAliases needs
  to be supported

- Only the FALSE value for attrsOnly needs to be supported

This subset provides a more general search capability. It is a
superset of the SearchRequest subset defined in Section 6.2.1. The
elements added to this service are:

- singleLevel and wholeSubtree scope needs to be supported

- sizeLimit is included

- timeLimit is included

- Enhanced filter capability

An application providing a LDAP repository search service MAY
implement other aspects of the SearchRequest as well.

7.2.2. Search Response

The full LDAPv2 SearchResponse is defined in RFC 1777.

An application providing a LDAP repository search service over
LDAPv2 MUST implement the full SearchResponse.

7.3. Unbind

An application providing a LDAP repository search service MUST
implement the full UnbindRequest.

8. LDAP Repository Modify

To add, delete and modify PKI information in a repository requires
a subset of the following LDAP operations:

```
  BindRequest (and BindResponse)
  ModifyRequest (and ModifyResponse)
  AddRequest (and AddResponse)
  DelRequest (and DelResponse
  UnbindRequest
```

The subset of each operation REQUIRED is given below.

8.1. Bind

The full LDAP v2 Bind Request is defined in RFC 1777.

An application providing a LDAP repository modify service MUST
implement the following subset of this operation:

```
  BindRequest ::=
    [APPLICATION 0] SEQUENCE {
       version     INTEGER (2),
       name        LDAPDN,
       simpleauth [0] OCTET STRING
       }
```

A LDAP repository modify service MUST implement authenticated
access.

The BindResponse subsets needed are the same as those described in

Section 6.1.2.

8.2. Modify

8.2.1. Modify Request

The full LDAPv2 ModifyRequest is defined in RFC 1777.

An application providing a LDAP repository modify service MUST
implement the following subset of the ModifyRequest protocol unit.

```
    ModifyRequest ::=
      [APPLICATION 6] SEQUENCE {
     object          LDAPDN,
     modification    SEQUENCE OF SEQUENCE {
                          operation     ENUMERATED {
                                            add     (0),
                                            delete  (1)
                                        },
                          modification  SEQUENCE {
                                        type  AttributeType,
                                        values SET OF
                                                AttributeValue
                                        }
                     }
         }
```

All aspects of the ModifyRequest MUST be supported, except for the
following:

- Only the add and delete values of operation need to be supported

8.2.2. Modify Response

The full LDAPv2 ModifyResponse is defined in RFC 1777.

An application providing a LDAP repository modify service MUST
implement the full ModifyResponse.

8.3. Add

8.3.1. Add Request

The full LDAPv2 AddRequest is defined in RFC 1777.

An application providing a LDAP repository modify service MUST
implement the full AddRequest.

8.3.2. Add Response

The full LDAPv2 AddResponse is defined in RFC 1777.

An application providing a LDAP repository modify service MUST implement the full AddResponse.

8.4. Delete

8.4.1. Delete Request

The full LDAPv2 DelRequest is defined in RFC 1777.

An application providing a LDAP repository modify service MUST implement the full DelRequest.

8.4.2. Delete Response

The full LDAPv2 DelResponse is defined in RFC 1777.

An application providing a LDAP repository modify service MUST implement the full DelResponse.

8.5. Unbind

An application providing a LDAP repository modify service MUST implement the full UnbindRequest.

9. Non-standard attribute value encodings

When conveyed in LDAP requests and results, attributes defined in X.500 are to be encoded using string representations defined in RFC 1778, The String Representation of Standard Attribute Syntaxes. These string encodings were based on the attribute definitions from X.500(1988). Thus, the string representations of the PKI informa- tion elements are for version 1 certificates and version 1 revoca- tion lists. Since this specification uses version 3 certificates and version 2 revocation lists, as defined in X.509(1997), the RFC 1778 string encoding of these attributes is inappropriate.

For this reason, these attributes MUST be encoded using a syntax similar to the syntax "Undefined" from section 2.1 of RFC 1778: values of these attributes are encoded as if they were values of type "OCTET STRING", with the string value of the encoding being the DER-encoding of the value itself. For example, when writing a userCertificate to the repository, the CA generates a DER-encoding of the certificate and uses that encoding as the value of the user- Certificate attribute in the LDAP Modify request.This encoding

style is consistent with the encoding scheme proposed for LDAPv3,
which is now being defined within the IETF.

Note that certificates and revocation lists will be transferred
using this mechanism rather than the string encodings in RFC 1778
and client systems which do not understand this encoding may
experience problems with these attributes.

10. Transport

An application providing a LDAP repository read service, LDAP repo-
sitory search service, or LDAP repository modify service MUST sup-
port LDAPv2 transport over TCP, as defined in Section 3.1 of RFC
1777.

An application providing a LDAP repository read service, LDAP repo-
sitory search service, or LDAP repository modify service MAY sup-
port LDAPv2 transport over other reliable transports as well.

11. Security Considerations

Since the elements of information which are key to the PKI service
(certificates and CRLs) are both digitally signed pieces of infor-
mation, additional integrity service is NOT REQUIRED. As neither
information element need be kept secret and anonymous access to
such information, for retrieval purposes is generally acceptable,
privacy service is NOT REQUIRED for information retrieval requests.

CAs have additional requirements, including modification of PKI
information. Simple authentication alone is not sufficient for
these purposes. It is RECOMMENDED that some stronger means of
authentication and/or (if simple authentication is used) some means
of protecting the privacy of the password is used, (e.g. accept
modifications only via physically secure networks, use IPsec, use
SSH or TLS or SSL tunnel). Without such authentication, it is pos-
sible that a denial-of-service attack could occur where the
attacker replaces valid certificates with bogus ones.

For the LDAP repository modify service, profiled in section 8,
there are some specific security considerations with respect to
access control. These controls apply to a repository which is under
the same management control as the CA. Organizations operating
directories are NOT REQUIRED to provide external CAs access permis-
sion to their directories.

The CA MUST have access control permissions allowing it to:

  For CA entries:

                - add, modify and delete all PKI attributes for its
                  own directory entry;
                - add, modify and delete all values of these attributes.

            For CRL distribution point entries (if used):
                - create, modify and delete entries of object class
                  cRLDistributionPoint immediately subordinate to its own
                  entry;
                - add, modify and delete all attributes, and all values of
                  these attributes for these entries.

            For subscriber (end-entity) entries:
                - add, modify and delete the attribute userCertificate and
                  all values of that attribute, issued by this CA
                  to/from these entries.

         The CA is the ONLY entity with these permissions.

      An application providing LDAP repository read, LDAP repository
      search, or LDAP repository modify service as defined in this
      specification is NOT REQUIRED to implement any additional security
      features other than those described herein, however an implementa-
      tion SHOULD do so.

[12]. References

[1] Lightweight Directory Access Protocol. Y. Yeong, T. Howes, S.
    Kille, RFC 1777, March 1995.

[2] The String Representation of Standard Attribute Syntaxes. T.
     Howes, S. Kille, W. Yeong, C. Robbins, RFC 1778, March 1995.

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

[13]. Author's Address

   Sharon Boeyen
   Entrust Technologies Limited
   750 Heron Road
   Ottawa, Ontario
   Canada K1V 1A7
   boeyen@entrust.com

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

    USA
    howes@netscape.com

    Patrick Richard
    Xcert Software Inc.
    Suite 1001, 701 W. Georgia Street
    P.O. Box 10145
    Pacific Centre
    Vancouver, B.C.
    Canada V7Y 1C6
    patr@xcert.com