INTERNET-DRAFT
Working Group
Intended Category: Standards Track
Expires 30 Nov 2002

Donna SkibbieKerberos
Jonathan Trostle
John Griffith
LIST OF OTHER AUTHORS
TO BE DONE
30 May 2002

Kerberos KDC LDAP Schema
draft-skibbie-krb-kdc-ldap-schema-02.txt

1. Status Of This Memo

2. Abstract

This document defines a schema for storing attributes used by implementations of Kerberos Version 5 Key Distribution Center (KDC) service in a directory that implements the Lightweight Directory Access Protocol (LDAP) Version 3. The directory must implement the LDAP Version 3 protocol as defined in RFC 2251 [2], RFC 2252 [3], RFC 2253 [4], RFC 2256 [5], 2829 [6], and 2830[7]. The schema defined in this document is referred to as the "KDC LDAP schema."

The KDC LDAP schema includes definitions for attributes defining a realm, a realm policy, principals, and principal policies. The KDC LDAP schema does not include definitions for attributes used to store keys.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

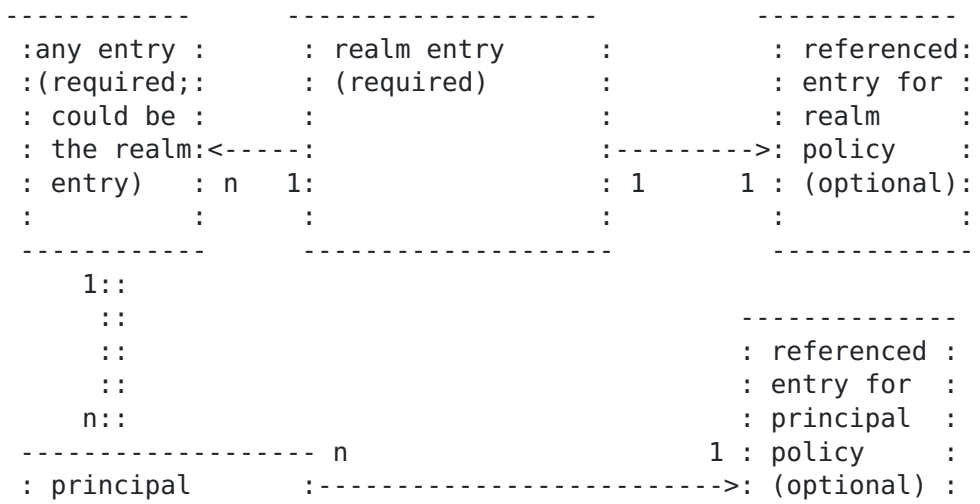
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [8].

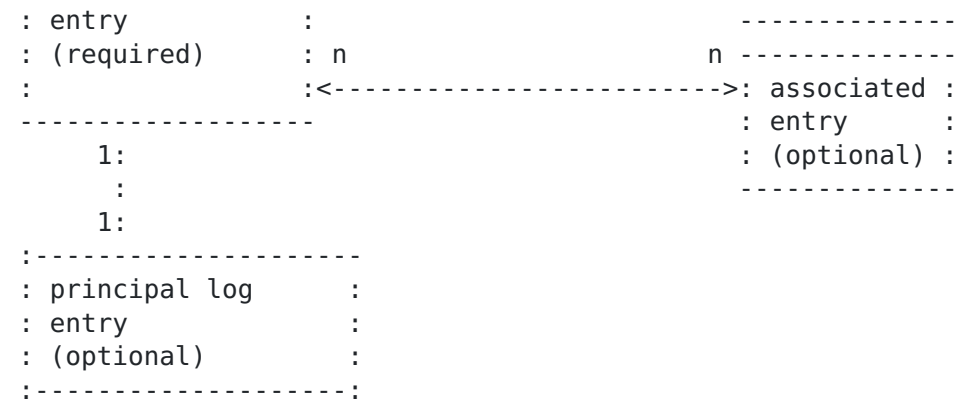
3. Overview

The KDC LDAP schema allows a KDC database to be stored in an LDAP back-end database (referred to as an LDAP directory) and accessed using standard LDAP interfaces. The benefits of doing this are as follows:
* Standard administration interface. The KDC LDAP schema together with the standard LDAP interfaces provide a standard interface that administrators and administration tools can use to configure KDC data.
* Inter-operation between KDCs at the database level. The KDC LDAP schema together with the standard LDAP interfaces allow different KDC implementations to be configured in the same realm with the different KDC implementations using the same KDC database.
* Sharing common security attributes with non-KDC applications. The standard LDAP interfaces allow KDC implementations to share common security attributes, such as names and password policy data, with non-KDC applications. This, of course, is provided that the administrator has configured the LDAP directory so that the common security attributes can be shared.
* Leveraging LDAP administration tools. Administrators can make use of existing LDAP administration tools to administer the KDC database.

The design goals of the KDC LDAP schema are as follows:
* use LDAP attribute definitions defined in RFC 2252, RFC 2256, and existing LDAP implementations, and provide a way that these attributes can be shared with non-KDC applications.
* provide a way of protecting sensitive KDC data stored in the LDAP directory.
* provide a way of configuring the KDC LDAP schema for optimum performance in accessing KDC data stored in the LDAP directory.
allow existing KDC attributes to be migrated easily to the LDAP directory.

The following figure illustrates the KDC LDAP schema:





The figure uses the following notations:

* Each box represents an LDAP directory entry.

* The vertical line between the principal entry and the principal log entry indicates a parent-child directory information tree (DIT) association between the DNs of these two entries. (For example, if the DN of the principal entry is "cn=Alice Smith, cn=Managers, ou=Austin", the DN of the principal log entry could be "cn=KrbLog, cn=Alice Smith, cn=Managers, ou=Austin".)

* The double vertical line between the entry labeled "any entry" and the principal entry represents an ancestor-child DIT association between the DNs of these two entries. (For example, if the DN of the entry labeled "any entry" is "ou=Austin", the DN of the principal entry could be "cn=Alice Smith, cn=Managers, ou=Austin".)

* An arrow between two entries represent a DN reference between the two entries. (For example, the arrow from the principal entry to the referenced entry for principal policy indicates that the principal entry could be configured with a reference to the DN of the referenced entry for principal policy.)

3.1 Realm Entry (Required)

The LDAP directory MUST contain an entry to represent the realm. This entry is referred to as the "realm entry."

The realm entry MAY be configured using the following object classes:

* KrbRealm structural object class
* KrbRealmExt auxiliary object class
* KrbPolicy auxiliary object class

The RDN of the realm entry MUST be "krbRealmName=<realm_name>", where realm name is the name of the realm. There are no more restrictions on the DN of the realm entry.

The realm entry MUST contain each of the following attributes, which provide information about the realm:

* krbRealmName--The name of the realm, which must be the same as the realm name specified in the RDN of the realm entry.
* krbPrincSubTree--The DN of each entry representing a sub-tree under which principals in the realm reside. This attribute allows the identity who configures the realm entry to indicate which sub-trees can be trusted to contain entries defining principals in the realm. Note that if principals can reside under the realm entry, the krbPrincSubTree attribute MUST contain the name of the realm entry.
* krbKdcServiceObject--The DN of each entry representing a KDC service in the realm.

The realm entry MAY contain one or more of the following additional attributes, which provide additional information about the realm:

* krbAdmAclDB
* krbAdmServiceObject
* krbEncTypeSupport
* * krbKeyType
* krbLogCfg
* krbPolicyObject
* krbPwdServiceObject
* krbRedundancyPolicy
* krbSaltTypeSupport
* krbTrustedAdmObject

Also, the realm entry MAY contain one or more of the following policy attributes, which provide information about the policy used in the realm:

* maxRenewAge *
* maxTicketAge *
* krbMultKeyVersionsOK *
* passwordExpireTime
* passwordDictFiles
* passwordMaxAge or maxPwdAge*
* passwordMinAge or minPwdAge*
* passwordMinDiffChars *
* passwordMinLength or minPwdLength*
* pwdHistoryLength *

3.2 Referenced Entry for Realm Policy Attributes (Optional)

The realm entry MAY contain a krbPolicyObject attribute that references the DN of another entry. The "referenced entry" (the entry referred to by the krbPolicyObject attribute) MAY be configured using the KrbPolicy auxiliary object class.

The referenced entry MAY contain any policy attributes that also MAY reside in the realm entry. (See the previous section for a list of the

policy attributes that MAY be in the realm entry.) If the same policy attribute resides in both the realm entry and the referenced entry, the policy attribute in the realm entry MUST take precedence.

There are no restrictions on the DN or RDN of the referenced entry. There are no required attributes in the referenced entry.

3.3 Principal Entries (Required)

The LDAP directory MUST contain one entry for each principal in the realm. Each of these entries is referred to as a "principal entry." A principal entry MAY be configured using either or both of the following object classes:

* KrbPrincipal auxiliary object class
* KrbPolicy auxiliary object class

Each principal entry MUST represent only one principal. A principal entry
MUST have a DN that locates the principal entry under a sub-tree listed in the krbPrincSubtree attribute of the realm entry. There are no restrictions on the RDN of a principal entry.

A principal entry MUST contain a krbPrincipalName attribute. This attribute MUST define a Kerberos principal identity in the format "<principal>@<realm>", where <principal> is the name of the principal and <realm> is the name of the realm.   The Kerberos principal identity MUST be unique within the realm.

A principal entry MAY contain one or more of the following attributes, which provide additional information about the principal:

* krbCurKeyVersion
* krbExtraData
* krbPolicyObject
* krbPrincipalType
* krbTaggedDataList
* pwdLastSet

Also, a principal entry MAY contain one or more of the following policy attributes, which provide information about the policy for the principal:

* accountExpires
* krbAttributes
* maxPwdAge
* maxRenewAge
* maxTicketAge
* minPwdAge
* minPwdLength
* krbMultKeyVersionsOK
* passwordExpireTime

* passwordDictFiles
* passwordMaxAge
* passwordMinAge
* passwordMinDiffChars
* passwordMinLength
* pwdHistoryLength
* secAcctExpires
* secAcctValid
* userAccountControl

3.4 Entries Associated with Principal Entries (Optional)

The schema provides an optional way of associating a principal entry with another entry through the use of aliases. This association is ignored by the KDC, but can be used by higher-level applications to associate a principal with a target entry and to verify that the target entry accepts this association.

There are three reasons why it might be necessary to configure alias associations. One reason is to allow an entry already configured with a principal identity to be associated with other principal identities. Another reason is to allow an entry configured in a remote part of the directory to be associated with a principal identity configured in a local part of the directory. A third reason is to allow an entry configured in a less secure part of the directory to be associated with a principal identity configured in more secure part of the directory.

The association MUST be as follows:

* the principal entry MUST contain a krbAliasedObjectName that references the target entry. This configuration MAY be done using the KrbAlias auxiliary object class.

* the target entry MUST contain a krbHintAliases attribute that references the principal entry. This configuration MAY be done using the KrbAlias auxiliary object class.

3.5 Referenced Entries for Principal Policy Attributes (Optional)

A principal entry MAY contain a krbPolicyObject attribute that references another entry. The "referenced entry" (the entry referenced by the krbPolicyObject attribute) MAY be configured using the KrbPolicy auxiliary object class.

The referenced entry MAY contain any policy attributes that also MAY reside in the principal entry. (See the previous section for a list of policy attributes that MAY reside in the a principal entry.) If the same policy attribute exists in both the principal entry and the referenced entry, the attribute in the referenced entry MUST take precedence.

There are no restrictions on the DN or RDN of the referenced entry.
There are no required attributes in the referenced entry.

3.6 Principal Log Entries (Optional)

The realm entry MAY have a krbLogCfg attribute that contains a value of TRUE. If so, the LDAP directory MUST contain an entry that represents the log-in activity record of each principal. Each of these entries is referred to as a "principal log entry." A principal log entry MAY be configured using the KrbLog structural object class.

The DN of a principal log entry MUST logically locate the entry in the LDAP DIT directly below the associated principal entry. The RDN of a principal log entry must be "cn=KrbLog". The creator of a principal log entry MUST be an identity that is listed in either the krbKdcServiceObject attribute or the krbTrustedAdmObject attribute of the realm entry.

A principal log entry MUST contain the following attributes:

* badPasswordTime--The last time an unsuccessful log-in was attempted using an incorrect password.
* badPwdCount--The number of incorrect log-in attempts using incorrect passwords.
* lastLogon--The last time a successful authentication was performed.

4. Syntaxes

The KDC LDAP schema uses the following syntaxes in attribute type definitions:

* Syntaxes listed in RFC 2252

* The interval syntax

The interval syntax is defined in the Microsoft Active Directory schema. The definition is as follows:

```
(
1.2.840.113556.1.4.906
NAME 'Interval'
DESC 'Large integer.  Use for 64-bit values.'
)
```

5. Attribute Types

The KDC LDAP schema uses the attribute types listed in this section and RFC 2256.

5.1 New Attribute Types

```
(
1.3.18.0.2.4.1899
NAME 'krbAdmAclDB'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'The location of an ACL database for a Kerberos
administration services,  The location must be specified as in
URL format; i.e., FILE://path/filename.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.1901
NAME 'krbAdmKeyLocation'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'The location of a keytab file containing the key used by
the Kerberos administration services,  The location must be
specified as in URL format; i.e., FILE://path/filename.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.1909
NAME 'krbAdmServiceObject'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of references to entries, with each entry
representing a Kerberos administration service in the realm.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1088
NAME 'krbAliasedObjectName'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
SINGLE-VALUE
DESC 'Forward reference to the entry for which this entry is an
alias.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1890
NAME 'krbAttributes'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'A value containing one or more flags.  The following flags
are available:
KRB5_KDB_NEW_PRINC = 0x00008000
```

KRB5_KDB_PWCHANGE_SERVICE = 0x00002000
KRB5_KDB_REQUIRES_HW_AUTH = 0x00000100
KRB5_KDB_REQUIRES_PWCHANGE = 0x00000200
KRB5_KDB_SUPPORT_DESMD5 = 0x00004000
KRB5_KDB_DISALLOW_DUP_SKEY = 0x00000020
KRB5_KDB_DISALLOW_POSTDATED = 0x00000001
KRB5_KDB_DIALLOW_PROXIABLE = 0x00000010
KRB5_KDB_DISALLOW_RENEWABLE = 0x00000008
KRB5_KDB_DIALLOW_TGT_BASED = 0x00000004
USER_TO_USER = 0x00010000
KRB5_KDB_DISALLOW_SVR = 0x00001000
DISALLOW_FORWARDABLE = 0x00001001
REQUIRES_PRE_AUTH = 0x00001002
ALL_TIX = 00001004
DELEGATE_OK = 00001008'
EQUALITY integerMatch
)

  (
1.3.18.0.2.4.1898
NAME 'krbCurKeyVersion'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'A value indicating the current version of a key.'
)

(
TODO: CHECK ON THIS WITH RFC1510 AND CRYTO DRAFT (SABU)
1.3.18.0.2.4.1892
NAME 'krbEncTypeSupport'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'A set of supported encryption type values. See krbKeyType
for encryption type values. The available values are:
  ENCTYPE_NULL 00
  ENCTYPE_DES_CBC_CRC 01 (DES cbc mode with CRC-32 as defined in RFC
1510 [9])
  **ENCTYPE_DES_CBC_MD4 02 (DES cbc mode with RSA-MD4 as defined in RFC
1510 [9] and work-in-progress draft [10])**
  **ENCTYPE_DES_CBC_MD5 03 (DES cbc mode with RSA-MD5 as defined in RFC
1510 [9] and work-in-progress draft [10])**
  ENCTYPE_DES_CBC_HMAC_SHA1 (Triple DES cbc mode with SHA1/HMAC as
defined in work-in-progress draft [10])
 cbc mode raw)

  ENCTYPE_DES_CBC_MOD_CRC 12 (DES cbc mode with modified CRC-32 as
defined in work-in-progress draft [10].
  ENCTYPE_RSA_PRIVKEY 91 (RSA private key; required for
support of DCE)
  ENCTYPE_UNKNOWN 99'
 EQUALITY integerMatch
)

(
1.3.18.0.2.4.1911
NAME 'krbExtraData'

SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'Extra data that is associated with a Kerberos principal and
that has an application-specific meaning. This attribute is
provided to support the Kerberos kadmin APIs.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.1154NAME 'krbHintAliases'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of backward references to entries that can serve as
aliases for this entry.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1907
NAME 'krbKdcServiceObject'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of references to entries, with each entry
representing a KDC service in the realm.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1888
NAME 'krbKeyType'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (IA5String)
DESC 'A set of key types. Each key type is specified in
the following format:
0 1 2 3 4
+--+--+--+--+
| enc |salt |
|type |type |
+--+--+--+--+
where
"enc type" is two decimal characters indicating the encryption
type of the key. See krbEncTypeSupport for a list of available
encryption type values.
"salt type" is two decimal characters indicating
the salt type of the key. See krbSaltTypeSupport for a list of
available salt type values.
For example, "0199" indicates a key that is generated with DES
encryption and no salt. As another example, "0500" indicates that
a key that is generated using triple DES encryption and a normal
salt type value.'
EQUALITY caseIgnoreMatch
)

(
krbLogCfg-oid
NAME 'krbLogCfg'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 (boolean)

```
SINGLE-VALUE
DESC 'True if the LDAP directory will contain a log entry for each
principal in the realm.'
EQUALITY booleanMatch
)


 (
1.3.18.0.2.4.1884
NAME 'krbMultKeyVersionsOK'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 (boolean)
SINGLE-VALUE
DESC 'True if multiple versions of a key for each encryption type
can be stored for this account.'
EQUALITY booleanMatch
)

(
1.3.18.0.2.4.1881
NAME 'krbPolicyName'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'Name for a Kerberos policy in the form <policy>@<realm>.
<policy> is the name of a policy and must be unique within
the realm.  <realm> is the name of the realm.  The realm
name must conform to the rules described in RFC 1510 and
must be the same as the realm name specified in the
krbRealmName attribute of the realm entry.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.1904
NAME 'krbPolicyObject'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
SINGLE-VALUE
DESC 'Forward reference to an entry containing policy
information.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1091
NAME 'krbPrincipalName'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'Kerberos principal identity for a user in the form
<principal>@<realm>. <principal> is the name of the principal
and must conform to the rules described in RFC 1510.  <realm> is
the realm name.  The realm name must conform to the rules
described in RFC 1510 and must be the same as the realm name
specified in the krbRealmName attribute of the realm entry.'
EQUALITY caseExactMatch
)
```

```
(
1.3.18.0.2.4.1883
NAME 'krbPrincipalType'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'Value defining the type of a principal.  The available
principal type values are:
0 = KRB5_NT_UNKNOWN
1 = KRB5_NT_PRINCIPAL
2 = KRB5_NT_SRV_INST
3 = KRB5_NT_SRV_HST
4 = KRB5_NT_SRV_XHST
5 = KRB5_NT_UID'
)

(
1.3.18.0.2.4.1156
NAME 'krbPrincSubTree
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of forward references to an entry that starts a sub-
tree where principals in the realm are configured.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1902
NAME 'krbPwdServiceObject'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of references to entries, with each entry
representing a password service in the realm.  This attribute is needed
only if the realm uses a password service that is different from the
service specified in the krbTrustedAdmObject attribute of the realm
entry.'
EQUALITY dnMatch
)

(
1.3.18.0.2.4.1157
NAME 'krbRealmName'
SYNTAX 1.3.6. 11.4.1.1466.115.121.1.15 (directory string)
SINGLE-VALUE
DESC 'Name of a security realm.  The realm name must
conform to the rules listed in RFC 1510.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.1885
NAME 'krbRedundancyPolicy'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'One of the following values indicating which set of
attributes to use for those attributes that have the same logical
```

meaning:
01 -- Use the set of attributes from the Netscape or IBM/Tivoli schema (default)
02 -- Use the set of attributes from the Microsoft schema. The following table lists the sets of attributes that have the same logical meanings and the schema's in which these attributes are defined:

Netscape or IBM/Tivoli Schema	Microsoft Schema
passwordExpireTime	computed from pwdLastSet and maxPwdAge
passwordMaxAge	maxPwdAge
passwordMinAge	minPwdAge
passwordMinLength	minPwdLength
secAcctExpires	accountExpires
secAcctValid	userAccountControl (!ACCOUNT_DISABLE)'

)

```
(
1.3.18.0.2.4.1891
NAME 'krbSaltTypeSupport'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'A set of values defining the supported salt types.  See the
krbKeyType attribute for a list of salt types. The available
values are:
  KRB5_KDB_SALTTYPE_NORMAL = 0
  KRB5_KDB_SALTTYPE_V4 = 1
  KRB5_KDB_SALTTYPE_NOREALM = 2
  KRB5_KDB_SALTTYPE_ONLYREALM = 3
  KRB5_KDB_SALTTYPE_SPECIAL = 4
  KRB5_KDB_SALTTYPE_AFS3 = 5
  KRB5_KDB_NO_SALT_VALUE = 99'
EQUALITY integerMatch
)

(
 1.3.18.0.2.4.1893
NAME 'krbTaggedDataList'
SYNTAX  1.3.6.1.4.1.1466.155.121.1.40 (octet string)
DESC 'Set of tagged data structures that is associated with a
Kerberos principal and that is defined by a Kerberos kadmin
application.  This attribute is provided to support the Kerberos
kadmin APIs.'
EQUALITY octetStringMatch
)

(
1.3.18.0.2.4.1903
NAME 'krbTrustedAdmObject'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.12 (DN)
DESC 'A set of forward references to trusted administration tools.'
```

```
EQUALITY dnMatch
)
```

5.2 Attribute Types Defined in the Netscape Schema

```
(
2.16.840.1.113730.3.1.97
NAME 'passwordMaxAge'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'Specifies, in seconds, the period of time passwords can be
used before they expire.'
)

(
1.3.18.0.2.4.465
NAME 'passwordMinAge'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'Specifies, in seconds, the period of time a password must
be in effect before a user can change it.'
)

(
2.16.840.1.113730.3.1.99
NAME 'passwordMinLength'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'Specifies the minimum number of characters required for a
user's password.'
)
```

5.3 Attribute Types Defined in the Microsoft Active Directory Schema

```
(
1.2.840.113556.1.4.159
NAME 'accountExpires'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating when the account will expire. The value
is stored as a large integer that represents the number of seconds
elapsed since 00:00:00, January 1, 1970. A value of TIMEQ_FOREVER
(-1) indicates that the account never expires.'
)

(
1.2.840.113556.1.4.49
NAME 'badPasswordTime'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating the last time the user tried to log onto
the account using an incorrect password.  The value is stored as
```

```
a large integer that represents the number of seconds elapsed
since 00:00:00, January 1, 1970. A value of zero (0) means the
last password time is unknown.'
)

(
1.2.840.113556.1.4.12
NAME 'badPwdCount'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'A value indicating the number of times the user tried
to log on to the account using an incorrect password.
A value of zero (0) indicates that the value is unknown.
'
)

(
1.2.840.113556.1.4.52
NAME 'lastLogon'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating when the last logon occurred.
The value is stored as a large integer that represents
the number of seconds elapsed since 00:00:00, January 1, 1970.
A value of zero (0) means that the last logon time is
unknown.'

)

(
1.2.840.113556.1.4.95
NAME 'pwdHistoryLength'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'A value indicating the number of previous passwords saved
in the history list. The user cannot reuse a password that is in
the history list.'
)

(
1.2.840.113556.1.4.96
NAME 'pwdLastSet'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating when the user last set the password.
The value is stored as a large integer that represents
the number of seconds elapsed since 00:00:00, January 1, 1970.
The system uses the value of this property and the maxPwdAge
property of the domain containing the user object to calculate
the password expiration date (sum of pwdLastSet for the user and
maxPwdAge of the user's domain).'
)

(
```

```
1.2.840.113556.1.4.74
NAME 'maxPwdAge'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating the maximum amount of time, in seconds,
after which the password must be changed by the owner.'
)

(
1.2.840.113556.1.4.75
NAME 'maxRenewAge'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'Value indicating the maximum renewable lifetime, in seconds,
of a Kerberos ticket.'
)

(
1.2.840.113556.1.4.77
NAME 'maxTicketAge'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating the maximum lifetime, in seconds,
of a Kerberos ticket.'
)

(
1.2.840.113556.1.4.78
NAME 'minPwdAge'
SYNTAX 1.2.840.113556.1.4.906 (interval)
SINGLE-VALUE
DESC 'A value indicating the minimum amount of time, in seconds,
before the password is allowed to be changed.'
)

(
1.2.840.113556.1.4.79
NAME 'minPwdLength'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'A value indicating the minimum number of characters that
must be typed in for a password.'
)

(
1.2.840.113556.1.4.8
NAME 'userAccountControl'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
SINGLE-VALUE
DESC 'A value containing one or more attributes that apply to an
account.  Each attribute is set with a flag.  Refer to the
Microsoft Active Directory documentation for a complete list of
flags.  The following flags are used in this KDC LDAP schema:
UF_ACCOUNT_DISABLE =  0x0001
```

```
UF_DONT_EXPIRE_PASSWD  =  0x10000
UF_TRUSTED_FOR_DELEGATION =  0x80000
UF_USE_DES_KEY_ONLY  =  0x200000
UF_DONT_REQUIRE_PREAUTH =  0x400000'
)
```

5.4 Attribute Types Defined in the IBM/Tivoli Schema

```
(
1.3.18.0.2.4.463
NAME 'passwordDictFiles'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 (directory string)
DESC 'Password dictionary files.'
EQUALITY caseExactMatch
)

(
1.3.18.0.2.4.485
NAME 'passwordExpireTime'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 (generalizedTime)
DESC ' Defines the date and time when
a user password expires.'
)

(
1.3.18.0.2.4.499
NAME 'passwordMinDiffChars'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 (integer)
DESC 'Specifies the minimum number of different (unique)
characters required for a user's password.'
)

(
1.3.6.1.4.1.4228.1.12
NAME 'secAcctExpires'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 (generalizedTime)
SINGLE-VALUE
DESC 'The date when a security account expires.'
)

(
1.3.6.1.4.1.4228.1.4
NAME 'secAcctValid'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 (boolean)
SINGLE-VALUE
DESC 'A boolean value indicating whether a security account is
valid.'
)
```

6. Object Classes

```
The KDC LDAP schema uses the object classes listed in this
section.

(
1.3.18.0.2.6.261
NAME 'KrbAlias'
DESC 'An auxiliary object class for use in configuring an
association between an entry containing security identity
information and another entry.'
SUP top
AUXILIARY
MAY (krbAliasedObjectName $ krbHintAliases)
)

(
1.3.18.0.2.6.364
NAME 'KrbLog'
DESC 'A structural object class for use in configuring an entry
to represent a Kerberos login activity record for an associated
Kerberos principal.'
SUP top
STRUCTURAL
MUST ( cn )
MAY ( badPasswordTime $ badPwdCount $ lastLogon)
)

(
1.3.18.0.2.6.363
NAME 'KrbPolicy'
DESC '  An auxiliary object class for use in configuring Kerberos
policy attributes for an associated Kerberos principal or
Kerberos realm.'
SUP top
AUXILIARY
MAY ( accountExpires $ krbAttributes $ krbPolicyName $ maxPwdAge
$ maxRenewAge $ maxTicketAge $ minPwdAge $ minPwdLength $
krbMultKeyVersionsOK $ passwordExpireTime $ passwordDictFiles
$ passwordMaxAge $ passwordMinAge $ passwordMinDiffChars $
passwordMinLength $ pwdHistoryLength $ secAcctExpires $
secAcctValid $ userAccountControl)
)

(
1.3.18.0.2.6.360
NAME 'KrbPrincipal'
DESC 'An auxiliary class for use in configuring an entry to
represent a Kerberos principal.'
SUP top
AUXILIARY
MUST (krbPrincipalName)
MAY (krbCurKeyVersion $
krbExtraData $
krbPolicyObject $ krbPrincipalType $ krbTaggedDataList $
```

```
pwdLastSet)
)

(
1.3.18.0.2.6.263
NAME 'KrbRealm'
DESC  A structural object class for use in configuring an entry
to represent a Kerberos realm.'
SUP top
STRUCTURAL
MUST ( krbPrincSubTree $ krbRealmName )
)

(
1.3.18.0.2.6.358
NAME 'KrbRealmExt'
DESC 'An auxiliary object class for use in configuring additional
attributes in an entry representing a Kerberos realm.'
SUP KrbPolicy
AUXILIARY
MAY ( krbAdmAclDB $ krbAdmServiceObject $ krbEncTypeSupport $
krbKdcServiceObject $ krbKeyType $ krbLogCfg $ krbPolicyObject $
krbPwdServiceObject $ krbRedundancyPolicy $
krbSaltTypeSupport $ krbTrustedAdmObject )
)
```

7. Examples of the Schema

The following are examples of entries defined in the KDC LDAP schema.

7.1. Example of a Realm Entry

The following is an example of a realm entry. In this example, all the realm policy attributes reside in the realm entry.

```
  dn: krbRealmName=Payroll, ou=Austin
  objectclass: KrbRealm
  objectclass: KrbRealmExt
  objectclass: KrbPolicy
  krbRealmName: Payroll
  krbPrincSubTree: cn=users, ou=Austin
  krbKdcServiceObject: serviceName=serverA, dc=payroll, ou=Austin
  krbKdcServiceObject: serviceName=serverB, dc=payroll, ou=Austin
  <additional KrbRealmExt attributes>
  <KrbPolicy attributes>
```

7.2. Example of a Referenced Entry for Realm Policy Attributes

The following is an example of a referenced entry for realm policy attributes. In this example, all the realm policy attributes reside in the referenced entry:

```
  dn: krbRealmName=Payroll, ou=Austin
  objectclass: KrbRealm
  objectclass: KrbRealmExt
  <KrbRealm and KrbRealmExt attributes>
  krbPolicyObject: cn=MyPolicy, ou=Austin

  dn: cn=MyPolicy, ou=Austin
  objectclass: PasswordPolicy
  <PasswordPolicy attributes>
  objectclass: KrbPolicy
  <KrbPolicy attributes>
```

7.3. Example of a Principal Entry

The following is an example of a principal entry. In this example, the principal entry was configured by adding the KrbPrincipal and KrbPolicy auxiliary object classes to an existing person entry:

```
  dn: cn=Alice Smith, cn=users, ou=Austin
  objectclass: Person
  cn: Alice Smith
  <additional Person attributes>
  objectclass: KrbPrincipal
  objectclass: KrbPolicy
  krbPrincipalName: alice@Payroll
  <additional principal attributes>
    <principal policy attributes>
```

7.4. Example of an Entry Associated with a Principal Entry

The following are two examples of an entry associated with a principal entry. In the first example, an existing Alice Smith person entry, which was configured as the principal identity of alice@Payroll, is associated with a second principal identity of alice@PURCHASING.

```
  dn: cn=Alice Smith, cn=users, ou=Austin
  objectclass: Person
  cn: Alice Smith
  <additional Person attributes>
  objectclass: KrbPrincipal
  krbPrincipalName: alice@Payroll
  <additional KrbPrincipal attributes>
  objectclass:  KrbAlias
  krbHintAliases:  cn=alice@PURCHASING, krbRealmName=PURCHASING,
ou=Austin

  dn: cn=alice@PURCHASING, krbRealmName=PURCHASING, ou=Austin
  objectclass: Person
  cn: alice@PURCHASING
  sn: alice@PURCHASING
  objectclass: KrbPrincipal
  krbPrincipalName: alice@PURCHASING
  <additional KrbPrincipal attributes>
  objectclass:  KrbAlias
```

```
  krbAliasedObjectName:  cn=Alice Smith, cn=users, ou=Austin
```

In the second example, an association is made between a principal entry for bob@Payroll and a person entry for Bob Jones that exists in a remote or less secure part of the directory:

```
  dn: cn=bob@Payroll, cn=users, ou=Austin
  objectclass: Person
  cn: bob@Payroll
  sn: bob@Payroll
  objectclass: KrbPrincipal
  objectclass: KrbAlias
  krbPrincipalName: bob@Payroll
  <additional KrbPrincipal attributes>
  krbAliasedObjectName:  cn=Bob Jones, ou=Raleigh

  dn: cn=Bob Jones, ou=Raleigh
  objectclass: Person
  objectclass: KrbAlias
  cn: Bob Jones
  <additional Person attributes>
  krbHintAliases: cn=bob@Payroll, cn=users, ou=Austin
```

7.5. Example of a Referenced Entry for Principal Policy Attributes

The following is an example of a referenced entry for principal policy attributes. In this example, all the principal policy attributes are configured in the referenced entry.

```
  dn: cn=Alice Smith, cn=users, ou=Austin
  objectclass: Person
  cn: Alice Smith
  <additional Person attributes>
  objectclass: KrbPrincipal
  krbPrincipalName: alice@Payroll
  <additional principal attributes>
  krbPolicyObject: cn=MyPolicy, ou=Austin

  dn: cn=MyPolicy, ou=Austin
  objectclass: PasswordPolicy
  objectclass: KrbPolicy
  <PasswordPolicy attributes>
    <principal policy attributes>
```

7.6. Example of a Principal Log Entry

The following is an example of a principal log entry.

```
  dn: cn=KrbLog, cn=Alice Smith, cn=users, ou=Austin
  objectclass: KrbLog
  lastLogon:  200202180600
  badPasswordTime: 0
  badPwdCount: 0
```

8. Security Considerations

AUTHENTICATION DISCLOSURE:

This document describes a directory access protocol that provides both read and update access. Update access and read access to sensitive information 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 considered 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 that are as strong or stronger than Kerberos are standardized, clients and servers written according to this specification which make make use of update functionality or the reading of private information are UNLIKELY TO INTER-OPERATE, or MAY INTER-OPERATE ONLY IF AUTHENTICATION IS REDUCED TO AN UNACCEPTABLY WEAK LEVEL. (RFC 2829 mandates that LDAP servers supporting authentication based on user ID and password implement the digest authentication protocol defined in RFC 2831 [8], but this mechanism is considered to be weaker than the Kerberos.)

Implementers are hereby discouraged from deploying LDAPv3 clients or servers that implement the update functionality or the reading of sensitive information until a Proposed Standard for a strong mandatory authentication mechanism in LDAPv3 has been approved and published as an RFC.

The following entities must be trusted to protect KDC attributes as described in this section:

* Administrators of the KDC LDAP schema
* KDC services that use the KDC LDAP schema
* LDAP client libraries used to access the KDC LDAP schema
* LDAP servers and backend databases with access to the KDC LDAP schema
* Administrators of LDAP servers and backend databases with access to KDC attributes

8.1. Security Considerations for Administrators of the KDC LDAP Schema

All administrators of the KDC LDAP schema must be trusted and are responsible for:

* Ensuring that KDC attributes are configured in LDAP locations that can be accessed only by LDAP servers that comply with the security considerations described in "Section 7.4. Security Considerations for LDAP Servers and Backend Databases with Access to the KDC LDAP Schema".

* If LDAP client libraries are used to access the attributes in the schema, ensuring that these libraries comply with the security considerations described in "Section 7.3. Security Considerations for LDAP Client Libraries Used to Access the KDC LDAP Schema"

* Ensuring that KDC attributes are transmitted securely to and from the LDAP server. If KDC attributes are transmitted over the network, they must be transmitted using a security protocol with client and server authentication and data integrity.

* Protecting the realm entry so that only trusted identities can modify, delete, or add attributes in the entry; only trusted identities can rename or delete the entry; only trusted identities can insert new entries under the entry; and only trusted identities can read the values in the krbKdcServiceObject, krbPwdServiceObject, and krbTrustedAdmObject attributes.

* Protecting each LDAP sub-tree referenced by krbPrincSubTree so that only trusted identities can add, modify, or delete KDC attributes residing under the sub-tree.

* Protecting each LDAP entry referenced by krbPolicyObject so that so that only trusted identities can add, modify, or delete attributes in the entry.

* Before retrieving attributes from a principal log entry, verifying that
the entry was created by a KDC service in the realm.

8.2. Security Considerations for KDC Servers that Use the KDC LDAP Schema

All KDC servers that use the KDC LDAP schema must be trusted and are responsible for:

* Using LDAP client routines that comply with the security considerations described in "Section 7.3. Security Considerations for LDAP Client Libraries Used to Access the KDC LDAP Schema."

* Transmitting KDC attributes securely to and from LDAP. If KDC attributes are transmitted over the network, they must be transmitted using a security protocol with client and server authentication and data integrity.

* When creating a principal log entry, protecting this entry so that only
a KDC service in the realm can modify, delete, and insert this entry; and only a KDC service or a trusted identity in the realm can delete or rename this entry.

* Before retrieving attributes from a principal log entry, verifying that
the entry was created by KDC service in the realm.

8.3. Security Considerations for LDAP Client Libraries Used to Access the KDC LDAP Schema

All LDAP client libraries used to access the KDC LDAP schema must be trusted and are responsible for protecting this information from other identities on the same machine.

If a trusted LDAP client library cannot be obtained, it would be possible to develop a trusted LDAP client library, which could be used by KDC servers, administrators of the KDC LDAP schema, and administrators of LDAP servers that access the KDC LDAP schema. The estimated lines of code required to develop such a library is included in the estimated lines of code required to develop the libraries used by a trusted LDAP server. (See the next section.)

8.4. Security Considerations for LDAP Servers and Backend Databases that Can Access the KDC LDAP Schema

All LDAP servers and backend databases of LDAP servers that have access to the KDC LDAP schema must be trusted and are responsible for:

* If remote access is supported, providing a security protocol for transmitting attributes over the network. The protocol must support client and server authentication and data integrity, and must be as strong or stronger than the Kerberos authentication protocol.

* Providing a way to protect attributes from unauthorized access.

* Providing a way to audit access to attributes.

* Replicating attributes only to other trusted LDAP servers and backend databases, and replicating these attributes in a secure manner. If KDC attributes are transmitted over the network to a replica, they must be transmitted using a security protocol with client and server authentication and data integrity.

* If Kerberos is used to authenticate the KDC to the LDAP server, then the LDAP server secret key may not be accessible over the network.

If it is impossible to obtain an LDAP server that meets the level of trust described in this section, it would be possible to develop a trusted LDAP server that reads and writes KDC attributes to a small trusted database, such as a database used by a legacy KDC. The estimated lines of code required to develop such an LDAP server is:

```
    Backend routines for storing KDC
    attributes in a trusted database:   4K of new code

    LDAP libraries and includes: 25K of ported/analyzed code from
                                 OpenLDAP source

    LDAP server (SLAPD):  12K of ported/analyzed code from
                          OpenLDAP source
    LDAP replication server (SLURPD):  2K of ported/analyzed code
                                       from OpenLDAP source
```

8.5. Security Considerations for Administrators that Manage LDAP Servers and Backend Database with Access to the KDC LDAP Schema

The administrator of each LDAP server and backend database with access to attributes in the KDC LDAP schema is responsible for:

* If LDAP client libraries are used to access the attributes in the schema, ensuring that these libraries comply with the security considerations described in "Section 7.3. Security Considerations for LDAP Client Libraries Used to Access the KDC LDAP Schema."

* Enabling auditing of LDAP servers and backend databases when required.

* Ensuring that the LDAP servers will not allow a client to

authenticate its identity to the LDAP server using an authentication protocol that is weaker than the Kerberos authentication protocol.

9. Acknowledgments

The authors wish to thank the members of The Open Group Directory Interoperability Forum for their contributions to this document.

10. Expiration Date

This draft expires November 30, 2002.

11. Bibliography

[1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[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 X.500 Directory Access Protocol (v3): Attribute Syntax Definitions", RFC 2252, December 1997.

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

[5] Wahl, M., "A Summary of the X.500(96) User Schema for use with LDAPv3", RFC 2256, December 1997.

[6] Wahl, M. Authentication Methods for LDAP, Request for Comments 2829, May 2000.

[7] Hodges, J., Morgan, R., and Wahl, M., "Lighweight Directory Access Protocol (v3): Extension for Transport Layer Security", RFC 2830, May 2000.

[8] Leach, P. and C. Newman, "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000.

[9] J. Kohl, C. Neuman. The Kerberos Network Authentication Service (V5), Request for Comments 1510.

[10] Raeburn, K. "Encryption and Checksum Specifications for Kerberos 5," draft-ietf-krb-wg-crypto-00.txt, January 2002.

12. Authors' Addresses

Donna Skibbie

IBM Corporation
1140 Burnet Road
Austin, TX 78758
Phone: (512) 838-3896
Email: donnas@us.ibm.com

Jonathan Trostle
ADDRESS TO BE DONE

John Griffith
Entegrity Solutions Corporation
32 DW Highway
Merrimack, NH 03054
Email: john.griffith@entegrity.com

OTHER AUTHORS' ADDRESSES
TO BE DONE

13. Open Issues

(1) Bob Joslin: This draft needs to allow an optional association between an entry and a password policy by means of placement in the schema, as defined in http://www.ietf.org/internet-drafts/draft-behera-ldap-password-policy-05.txt,.

(2) Bob Joslin: This draft needs to reference http://www.ietf.org/internet-drafts/draft-joslin-config-schema-03.txt for information on attribute mapping. (krbRedundancyPolicy not necessary.)

(3) Mortezza Anzani: Change the draft from the STANDARDS track to the INFORMATIONAL track.

(4) Mortezza Anzani: Move all attributes having to do with keys to the Keys Extension draft. (Comment from Donna Skibbie: All the attributes that are used for storing key data are defined in the KDC Keys Extension draft. This draft defines the following attributes, which are used to define which types of keys are supported in the realm: krbAdmKeyLocation, krbCurKeyVersion, krbEncTypeSupport, krbKeyType, and krbMultKeyVersionsOK. Since these attributes are not used to store key data, do we still want to move these attributes to the Keys Extension draft?)

(5) Sabu Shefeeq: I think it is worthwhile to do a security analysis and mention the consequences of not following the security precuations mentioned here.
For example:
What is the consequence of a password policy attribute "passwordExpireTime" being broken and being able to be read/modify by a hacker?
Will he be able to crack the password? Will be be able to access the data/system protected by Kerberos application?

I know it is a time consuming job. But it may be good to thing in those lines.