An information model for Kerberos version 5
draft-ietf-krb-wg-kdc-model-12

Abstract

   This document describes an information model for Kerberos version 5
   from the point of view of an administrative service. There is no
   standard for administrating a kerberos 5 KDC. This document
   describes the services exposed by an administrative interface to a
   KDC.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   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."

   This Internet-Draft will expire on December 2, 2012.

Copyright Notice

Table of Contents

1. Introduction

   The Kerberos version 5 authentication service described in [RFC4120]
   describes how a Key Distribution Center (KDC) provides authentication
   to clients. The standard does not stipulate how a KDC is managed and
   several "kadmin" servers have evolved. This document describes the
   services required to administer a KDC and the underlying information
   model assumed by a kadmin-type service.

   The information model is written in terms of "attributes" and
   "services" or "interfaces" but the use of these particular words must
   not be taken to imply any particular modeling paradigm. Neither an
   object oriented model nor an LDAP schema is intended. The author has
   attempted to describe in natural language the intended semantics and
   syntax of the components of the model. An LDAP schema (for instance)
   based on this model will be more precise in the expression of the
   syntax while preserving the semantics of this model.

   Implementations of this document MAY decide to change the names used
   (e.g. principalName). If so an implementation MUST provide a name to
   name mapping to this document. In particular schema languages may
   have different conventions for caseing, eg camelCase vs use of '_'
   and '-' to separate 'words' in a name. Implementations MUST call out
   such conventions explicitly.

   Implementations of this document MUST be able to support default
   values for attributes as well as the ability to specify syntax for
   attribute values.

[2](#). Requirements notation

   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 [[RFC2119](#)].

   This document describes an information model for kerberos 5 but does
   not directly describe any mapping onto a particular schema- or
   modelling language. Hence an implementation of this model consists
   of a mapping to such a language - e.g. an LDAP or SQL schema. The
   precise interpretation of terms from [[RFC2119](#)] therefore require some
   extra explanation.

   The terms MUST or REQUIRED means that schema implementing this model
   must have a way to represent the feature (i.e that it is mandatory to
   implement) but that unless otherwise specified the feature may
   represent an optional element in the chosen schema definition
   language.

   However MUST also means that a KDC or administrative interface
   implementing this information model MUST provide the feature and
   associated behavior consistent with schema.

   For instance, principalLastFailedAuthentication (cf below) represents
   the last time an authentication failed for a principal. In an LDAP
   schema (for instance) this may be represented as an optional
   attribute even though all KDCs implementing this specification must
   support this attribute.

   The terms MAY or OPTIONAL means that the feature is optional to
   implement by a KDC or administrative interface implementing this
   information model. MAY also means that the feature is optional to
   implement in schema.

   Implementors of schema should be aware that unless there is a way to
   represent critical but optional elements in the schema definition
   language confusion may arise when optional elements are used but not
   understood by all implementations in a particular deployment.

3. **Information model demarcation**

   The information model specified in the next chapter describes
   objects, properties of those objects and relations between those
   objects. These elements comprise an abstract view of the data
   represented in a KDC. It is important to understand that the
   information model is not a schema. In particular the way objects are
   compared for equality beyond that which is implied by the
   specification of a syntax is not part of this specification. Nor is
   ordering specified between elements of a particular syntax.

   Further work on Kerberos will undoubtedly prompt updates to this
   information model to reflect changes in the functions performed by
   the KDC. Such extensions to the information model should always use
   a normative reference to the relevant RFCs detailing the change in
   KDC function.

   This model describes a number of elements related to password policy
   management. Not all of the elements in this model are unique to
   Kerberos; an LDAP implementation of this model should incorporate
   existing LDAP schema where functional overlap exists, rather than
   defining additional Kerberos-specific elements.

4. Information model specification

4.1. Principal

   The fundamental entity stored in a KDC is the principal. The
   Principal is associated to keys and generalizes the "user" concept.
   The Principal MUST be implemented in full and MUST NOT be OPTIONAL in
   an implementation

4.1.1. Principal: Attributes

4.1.1.1. principalName

   The principalName MUST uniquely identify the Principal within the
   administrative context of the KDC. The principalName MUST be
   equivalent to the string representation of the Principal name
   (section 2.1.1 of [RFC1964]) including, if applicable for the name
   type, the realm.

   The attribute MAY be multi-valued if the implementation supports
   aliases and/or enterprise names. In that case exactly one of the
   principalName values MAY be designated the canonical principalName
   and if the implementation supports enctypes which require salt then
   exactly one of the values of principalName MAY be designated as the
   canonical salting principalName.

   Implementations (i.e. schema) that support enterprise names and/or
   aliases SHOULD provide for efficient lookup of Principal objects
   based on alias/enterprise name.

4.1.1.2. principalNotUsedBefore

   The Principal MUST not be used before this date. The syntax of the
   attribute MUST be Internet Date/Time Format from [RFC3339]. The
   attribute MUST be single-valued.

4.1.1.3. principalNotUsedAfter

   The Principal MUST not be used after this date. The syntax of the
   attribute MUST be Internet Date/Time Format from [RFC3339]. The
   attribute MUST be single-valued.

4.1.1.4. principalIsDisabled

   A boolean attribute used to disable a Principal. The attribute
   SHOULD default to boolean FALSE.

4.1.1.5. principalNumberOfFailedAuthenticationAttempts

   This single-valued integer attribute contains a count of the number
   of times an authentication attempt was unsuccessful for this
   Principal. Implementations SHOULD NOT allow this counter to be
   reset.

4.1.1.6. principalLastFailedAuthentication

   This single-valued attribute contains the time and date for the last
   failed authentication attempt for this Principal. The syntax of the
   attribute MUST be Internet Date/Time Format from [RFC3339].

4.1.1.7. principalLastSuccessfulAuthentication

   This single-valued attribute contains the time and date for the last
   successful authentication attempt for this principal. The syntax of
   the attribute MUST be Internet Date/Time Format from [RFC3339].

4.1.1.8. principalLastCredentialChangeTime

   This single-valued attribute contains the time of the last successful
   change of credential (e.g. password or private key) associated with
   this Principal. The syntax of the attribute MUST be Internet Date/
   Time Format from [RFC3339].

4.1.1.9. principalCreateTime

   This single-valued attribute contains the time and date when this
   Principal was created. The syntax of the attribute MUST be Internet
   Date/Time Format from [RFC3339].

4.1.1.10. principalModifyTime

   This single-valued attribute contains the time and date when this
   Principal was last modified excluding credentials change. The syntax
   of the attribute MUST be Internet Date/Time Format from [RFC3339].

4.1.1.11. principalMaximumTicketLifetime

   This single-valued attribute contains the time in seconds
   representing the maximum lifetime for tickets issued for this
   Principal.

4.1.1.12. principalMaximumRenewableTicketLifetime

   This single-valued attribute contains the delta time in seconds
   representing the maximum amount of time a ticket may be renewed for

this Principal.

4.1.1.13. principalAllowedEnctype

This OPTIONAL multi-valued attribute lists the enctypes allowed for
this principal. If empty or absent any enctype supported by the
implementation is allowed for this Principal.

This attribute is intended as a policy attribute and restricts all
uses of enctypes including server, client, and session keys. Data
models MAY choose to use policy objects in order to represent more
complex decision mechanisms.

4.1.2. Principal: Associations

Each Principal MAY be associated with 0 or more KeySet and MAY be
associated with 0 or more Policies. The KeySet is represented as an
object in this model since it has attributes associated with it (the
key version number). In typical situations the Principal is
associated with exactly 1 KeySet but implementations MUST NOT assume
this case, i.e. an implementation of this standard MUST be able to
handle the general case of multiple KeySet associated with each
principal. Multiple KeySets may for instance be useful when
performing a key rollover for a principal.

4.2. KeySet

In Kerberos principals are associated with zero or more symmetric
secret keys, and each key has a key version number (kvno) and
enctype. In this model we group keys by kvno into KeySet objects. A
Principal can have zero or more KeySet objects associated with it,
each of which MUST have one or more keys. Each KeySet is associated
with exactly one principal. Schemas derived from this model MAY lack
a direct analogue of KeySet as described in this document.

It is expected that most Kerberos implementations will use a special-
purpose interface for setting and changing Principal passwords and
keys.

If a server supports an enctype for a Principal that enctype must be
present in at least one key for the Principal in question. For any
given enctype a KeySet MUST NOT contain more than one Key with that
enctype.

The security of Kerberos 5 depends absolutely on the confidentiality
and integrity of the Key objects stored in the KDC. Implementations
of this standard MUST facilitate, to the extent possible, an
administrator's ability to place more restrictive access controls on

   KeySets than on other Principal data, and to arrange for more secure
   backup for KeySets.

4.2.1. KeySet: Attributes

4.2.1.1. kvno

   Also knowns as the key version number. This is a single-valued
   attribute containing a non-negative integer. This number is
   incremembed by one each time a key in the KeySet is changed.

4.2.2. KeySet: Associations

   To each KeySet MUST be associated a set of 1 or more Keys.

4.3. Key

   Implementations of this model MUST NOT REQUIRE keys to be
   represented.

4.3.1. Key: Attributes

4.3.1.1. keyEncryptionType

   The enctype SHOULD be represented as an enumeration of the enctypes
   supported by the KDC using the string name ("encryption type") of the
   enctype from the IANA registry of Kerberos Encryption Type Numbers.
   One example is 'aes128-cts-hmac-sha1-96'.

4.3.1.2. keyValue

   The binary representation of the key data. This MUST be a single-
   valued octet string.

4.3.1.3. keySaltValue

   The binary representation of the key salt. This MUST be a single-
   valued octet string.

4.3.1.4. keyStringToKeyParameter

   This MUST be a single-valued octet string representing an opaque
   parameter associated with the enctype. This parameter is specified
   in the "string-to-key" method in section 3 of [RFC3961].

4.3.1.5. keyNotUsedBefore

This key MUST NOT be used before this date. The syntax of the
attribute MUST be semantically equivalent with the standard ISO date
format. This MUST be a single-valued attribute.

4.3.1.6. keyNotUsedAfter

This key MUST NOT be used after this date. The syntax of the
attribute MUST be semantically equivalent with the standard ISO date
format. This MUST be a single-valued attribute.

4.3.1.7. keyIsDisabled

This is a boolean attribute which SHOULD be set to false by default.
If this attribute is true the key MUST NOT be used. This is used to
temporarily disable a key.

4.3.2. Key: Associations

None

4.3.3. Key: Remarks

The security of the keys is an absolute requirement for the operation
of Kerberos 5. If keys are implemented adequate protection from
unauthorized modification and disclosure MUST be available and
REQUIRED by the implementation.

4.4. Policy

Implementations SHOULD implement policy but MAY allow them to be
OPTIONAL. The Policy should be thought of as a 'typed hole'. i.e. an
opaque binary value paired with an identifier of type of data
contained in the binary value. Both attributes (type and value) must
be present.

4.4.1. Policy: Attributes

4.4.1.1. policyIdentifier

The policyIdentifier MUST be globally unique. Possible types of
identifiers include:

An Object Identifier (OID)

A URI

A UUID

The use of OIDs is RECOMMENDED for this purpose.

[4.4.1.2](#). **policyIsCritical**

This boolean attribute indicates that the KDC MUST be able to
correctly interpret and apply this policy for the Principal to be
used.

[4.4.1.3](#). **policyContent**

This is an optional single opaque binary value used to store a
representation of the policy. In general a policy cannot be fully
expressed using attribute-value pairs. The policyContent is OPTIONAL
in the sense that an implementation MAY use it to store an opaque
value for those policy-types which are not directly representable in
that implementation.

[4.4.1.4](#). **policyUse**

This is an optional single enumerated string value used to describe
the use of the policy. Implementations SHOULD provide this attribute
and MUST (if the attribute is implemented) describe the enumerated
set of possible values. The intent is that this attribute be useful
in providing an initial context-based filtering.

[4.4.2](#). **Mandatory-to-implement Policy**

All implementations that represent Policy objects MUST be able to
represent the policies listed in this section. Implementations are
not required to use the same underlying data-representation for the
policyContent binary value but SHOULD use the same OIDs as the
policyIdentifier. In general the expression of policy may require a
Turing-complete language. This specification does not attempt to
model policy expression language.

[4.4.2.1](#). **Password Quality Policy**

Password quality policy controls the requirements placed by the KDC
on new passwords. This policy SHOULD be identified by the OID
<TBD>.1.

[4.4.2.2](#). **Password Management Policy**

Password management policy controls how passwords are changed. This
policy SHOULD be identified by the OID <TBD>.2.

4.4.2.3. Keying Policy

   A keying policy specifies the association of enctypes with new
   principals, e.g. when a Principal is created one of the applicable
   keying policies is used to determine the set of keys to associate
   with the principal. This policy SHOULD be identified by the OID
   <TBD>.3.

4.4.2.4. Ticket Flag Policy

   A ticket flag policy specifies the ticket flags allowed for tickets
   issued for a principal. This policy SHOULD be identified by the OID
   <TBD>.4.

5. Implementation Scenarios

   There are several ways to implement an administrative service for
   Kerberos 5 based on this information model. In this section we list
   a few of them.

5.1. LDAP backend to KDC

   Given an LDAP schema implementation of this information model it
   would be possible to build an administrative service by back-ending
   the KDC to a directory server where principals and keys are stored.
   Using the security mechanisms available on the directory server keys
   are protected from access by anyone apart from the KDC.
   Administration of the principals, policy, and other non-key data is
   done through the directory server while the keys are modified using
   the set/change password protocol
   [I-D.ietf-krb-wg-kerberos-set-passwd].

5.2. LDAP frontend to KDC

   An alternative way to provide a directory interface to the KDC is to
   implement an LDAP-frontend to the KDC which exposes all non-key
   objects as entries and attributes. As in the example above all keys
   are modified using the set/change password protocol
   [I-D.ietf-krb-wg-kerberos-set-passwd]. In this scenario the
   implementation would typically not use a traditional LDAP
   implementation but treat LDAP as an access protocol to data in the
   native KDC database.

5.3. SOAP

   Given an XML schema implementation of this information model it would
   be possible to build a SOAP interface to the KDC. This demonstrates
   the value of creating an abstract information model which is mappable
   to multiple schema representations.

5.4. Netconf

   Given a YAML implementation of this information model it would be
   possible to create a Netconf-based interface to the KDC, enabling
   management of the KDC from standard network management applications.

6. Security Considerations

   This document describes an abstract information model for Kerberos 5.
   The Kerberos 5 protocol depends on the security of the keys stored in
   the KDC. The model described here assumes that keys MUST NOT be
   transported in the clear over the network and furthermore that keys
   are treated as write-only attributes that SHALL only be modified
   (using the administrative interface) by the change-password protocol
   [I-D.ietf-krb-wg-kerberos-set-passwd].

   Exposing the object model of a KDC typically implies that objects can
   be modified and/or deleted. In a KDC not all principals are created
   equal, so that for instance deleting krbtgt/EXAMPLE.COM@EXAMPLE.COM
   effectively disables the EXAMPLE.COM realm. Hence access control is
   paramount to the security of any implementation. This document does
   not mandate access control. This only implies that access control is
   beyond the scope of the standard information model, i.e. that access
   control may not be accessible via any protocol based on this model.
   If access control objects are exposed via an extension to this model
   the presence of access control may in itself provide points of attack
   by giving away information about principals with elevated rights etc.

7. IANA Considerations

   This document requires the allocation of several OIDs marked <TBD> in
   section 4.4.2 above. IANA should allocate a new arc under
   1.3.6.1.5.2.5 (iso.org.dod.internet.security.kerberosV5.policies)
   named "kdcPolicy" and assign each of the policy OIDs a new number
   under this arc.

8. Acknowledgments

The author wishes to extend his thanks to Love Hoernquist-Aestrand
<lha@kth.se> and Sam Hartman <hartmans@mit.edu> for their important
contributions to this document.

9. References

9.1. Normative References

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism",
           RFC 1964, June 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the
           Internet: Timestamps", RFC 3339, July 2002.

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for
           Kerberos 5", RFC 3961, February 2005.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The
           Kerberos Network Authentication Service (V5)", RFC 4120,
           July 2005.

9.2. Informative References

[I-D.ietf-krb-wg-kerberos-set-passwd]
           Williams, N., "Kerberos Set/Change Key/Password Protocol
           Version 2", draft-ietf-krb-wg-kerberos-set-passwd-08 (work
           in progress), November 2008.

Author's Address

     Leif Johansson
     Swedish University Network
     Thulegatan 11
     Stockholm

     Email: leifj@sunet.se
     URI: http://www.sunet.se