KERBEROS WORKING GROUP	Johansson	
Internet-Draft	SUNET	
Intended status: Standards Track	October 19, 2009	
Expires: April 22, 2010		

TOC

An information model for Kerberos version 5
draft-ietf-krb-wg-kdc-model-06

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.
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 documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.
This Internet-Draft will expire on April 22, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

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.

Table of Contents

1. Requirements notation
2. Introduction
3. How to interpret RFC2119 terms
4. Acknowledgments
5. Information model demarcation
6. Information model specification
    - 6.1. Principal
        - 6.1.1. Principal: Attributes
        - 6.1.2. Principal: Associations
    - 6.2. KeySet
        - 6.2.1. KeySet: Attributes
        - 6.2.2. KeySet: Associations
        - 6.2.3. KeySet: Remarks
    - 6.3. Key
        - 6.3.1. Key: Attributes
        - 6.3.2. Key: Associations
        - 6.3.3. Key: Remarks
    - 6.4. Policy
        - 6.4.1. Policy: Attributes
        - 6.4.2. Mandatory-to-implement Policy
7. Implementation Scenarios
    - 7.1. LDAP backend to KDC
    - 7.2. LDAP frontend to KDC
    - 7.3. SOAP
    - 7.4. Netconf
8. Security Considerations
9. IANA Considerations
10. References
    - 10.1. Normative References
    - 10.2. Informative References

§ Author's Address

---

1. Requirements notation

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

TOC

2. Introduction

The Kerberos version 5 authentication service described in [RFC4120] (Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)," July 2005.) describes how a Key Distribution Service (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 administrate 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 so that neither an object oriented model or 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 (eg principalName). If so an implementation MUST provide a name to name mapping to this document.

---

3. How to interpret RFC2119 terms

TOC

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 - eg an LDAP or SQL schema. The precise interpretation of terms from [RFC2119] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.) therefore require some extra explanation. The terms MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT mean that an implementation MUST provide a feature but does not mean that this feature MUST be REQUIRED by the implementation - eg an attribute is available in an LDAP schema but marked as OPTIONAL. If a feature must be implemented and REQUIRED this is made explicit in this model. The term MAY, OPTIONAL and RECOMMENDED means that an implementation MAY need to REQUIRE the feature due to the particular nature of the schema/modelling language. In some cases this is expressly forbidden by this model (feature X MUST NOT be REQUIRED by an implementation).

Note that any implementation of this model SHOULD be published as an RFC.

TOC

4. Acknowledgments

Love Hörnquist-Åstrand <lha@it.su.se> for important contributions.

---

5. Information model demarcation

TOC

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 MUST 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.

---

6. Information model specification

TOC

6.1. Principal

TOC

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

---

6.1.1. Principal: Attributes

TOC

6.1.1.1. principalName

TOC

The principalName MUST uniquely identify the principal within the administrative context of the KDC. The type of the principalName is not described in this document. It is a unique identifier and can be viewed as an opaque byte string which can be compared for equality.
The attribute MAY be multivalued if the implmementation supports aliases. In that case exactly one of the principalName values MUST be designated the canonical principalName and if the implementation supports enctypes which require salt then exactly one of the values of principalName MUST be designated as the canonical salting principalName.

6.1.1.2. principalNotUsedBefore

TOC

The principal may not be used before this date. The syntax of the attribute MUST be semantically equivalent with the standard ISO date format. The attribute MUST be single valued.

6.1.1.3. principalNotUsedAfter

TOC

The principal may not be used after this date. The syntax of the attribute MUST be semantically equivalent with the standard ISO date format. The attribute MUST be single valued.

6.1.1.4. principalIsDisabled

TOC

A boolean attribute used to (temporarily) disable a principal. The attribute SHOULD default to false.

6.1.1.5. principalNumberOfFailedAuthenticationAttempts

TOC

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.

6.1.1.6. principalLastFailedAuthentication

TOC

This single valued attribute contains the time and date for the last failed authentication attempt for this principal.

6.1.1.7. principalLastSuccessfulAuthentication

TOC

This single valued attribute contains the time and date for the last successful authentication attempt for this principal.

6.1.1.8. principalLastCredentialChangeTime

TOC

This single valued attribute contains the time and date for the last successful change of credential (eg password or private key) associated with this principal.

6.1.1.9. principalCreateTime

TOC

This single valued attribute contains the time and date when this principal was created

---

6.1.1.10. principalModifyTime

TOC

This single valued attribute contains the time and date when this principal was modified excluding credentials change.

6.1.1.11. principalMaximumTicketLifetime

TOC

This single valued attribute contains the delta time in seconds representing the maximum ticket lifetime for tickets issued for this principal.

TOC

6.1.1.12. principalMaximumRenewableTicketLifetime

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

6.1.1.13. principalAllowedEnctype

TOC

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.
Implementations MAY choose to use policy objects in order to represent more complex decision mechanisms.

6.1.1.14. principalRealm

TOC

This is a multi valued attribute listing the realms in which this principal exists using the string representation of the realm name(s).

6.1.2. Principal: Associations

TOC

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 (e.g an LDAP schema) MUST be able to handle the general case of multiple KeySet associated with each principal.

6.2. KeySet

TOC

A KeySet is a set of keys associated with exactly one principal. This object and its associations MUST NOT be REQUIRED by an implementation. It is expected that most implementations of this standard will use the set/change password protocol for all aspects of key management [I-D.ietf-krb-wg-kerberos-set-passwd] (Williams, N., “Kerberos Set/Change Key/Password Protocol Version 2,” November 2008.). This information model only includes these objects for the sake of completenes.

6.2.1. KeySet: Attributes

TOC

6.2.1.1. keySetVersionNumber

TOC

This is traditionally called the key version number (kvno). This is a single valued attribute containing a positive integer.

6.2.2. KeySet: Associations

TOC

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

6.2.3. KeySet: Remarks

TOC

The reason for separating the KeySet from the Principal is security. The security of Kerberos 5 depends absolutely on the security of the keys stored in the KDC. The KeySet type is provided to make this clear and to make separation of keys from other parts of the model clear. Implementations of this standard (eg an LDAP schema) MUST make a clear separation between the representation of KeySet from other information objects.

6.3. Key

TOC

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

6.3.1. Key: Attributes

TOC

TOC

6.3.1.1. keyEncryptionType

The enctype SHOULD be represented as an enumeration of the enctypes supported by the KDC.

6.3.1.2. keyValue

TOC

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

6.3.1.3. keySaltValue

TOC

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

6.3.1.4. keyStringToKeyParameter

TOC

This MUST be a single valued octet string representing an opaque parameter associated with the enctype.

6.3.1.5. keyNotUsedAfter

TOC

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.

6.3.1.6. keyNotUsedBefore

TOC

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.

TOC

6.3.1.7. 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.

6.3.2. Key: Associations TOC

None

6.3.3. Key: Remarks TOC

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.

6.4. Policy TOC

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.

6.4.1. Policy: Attributes TOC

6.4.1.1. policyIdentifier TOC

The policyIdentifier MUST be unique within the local administrative context and 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.

6.4.1.2. policyIsCritical

TOC

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

6.4.1.3. policyContent

TOC

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.

6.4.1.4. policyUse

TOC

This is an optional single enumerated string value used to describe the applicability of the policy. Implementations SHOULD provide this attribute and MUST (if the attribute is implemented) describe the enumerated set of possible values.

6.4.2. Mandatory-to-implement Policy

TOC

All implementations 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.

TOC

6.4.2.1. 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.

6.4.2.2. Password Management Policy

TOC

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

6.4.2.3. Keying Policy

TOC

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

6.4.2.4. Ticket Flag Policy

TOC

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.

---

7. Implementation Scenarios

TOC

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.

7.1. LDAP backend to KDC

TOC

Given an LDAP schema implementation of this information model it would be possible to build an administrative service by backending 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] (Williams, N., “Kerberos Set/Change Key/Password Protocol Version 2,” November 2008.).

7.2. LDAP frontend to KDC

TOC

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] (Williams, N., “Kerberos Set/Change Key/Password Protocol Version 2,” November 2008.). 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.

7.3. SOAP

TOC

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.

7.4. Netconf

TOC

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

---

8. Security Considerations

TOC

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] (Williams, N., “Kerberos Set/Change Key/Password Protocol Version 2,” November 2008.).
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 (at the time of writing - leifj) 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 is 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. etc.

---

9. IANA Considerations

TOC

This document requires the allocation of several OIDs marked <TBD> in the text.

---

10. References

TOC

10.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4120]	Neuman, C., Yu, T., Hartman, S., and K. Raeburn, “The Kerberos Network Authentication Service (V5),” RFC 4120, July 2005 (TXT).

10.2. Informative References

TOC

[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 (TXT).

Author's Address

TOC

	Leif Johansson
	Swedish University Network
	Thulegatan 11
	Stockholm
Email:	leifj@sunet.se
URI:	http://www.sunet.se