Internet Engineering Task Force (IETF)
Request for Comments: 6880
Category: Standards Track
ISSN: 2070-1721

L. Johansson
SUNET

March 2013

An Information Model for Kerberos Version 5

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 Key Distribution Center (KDC). This document describes the services exposed by an administrative interface to a KDC.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6880.

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. RFC 4120 does not stipulate how a KDC is managed, and several "kadmin" servers have evolved since RFC 4120 was written. 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 either "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 a Lightweight Directory Access Protocol (LDAP) [RFC4510] schema is intended. The author has attempted to describe, in prose, the intended semantics and syntax of the components of the model. For instance, an LDAP schema 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 typographical conventions, e.g., the use of an uppercase letter (as seen in camelCase) versus the 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 and have the ability to specify syntax for attribute values.

2. Requirements Notation

This document uses the standard key words ("MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL") that are defined in [RFC2119], but with modifications to those definitions as described below. The reason for this (which was discussed extensively in the Kerberos working group) is as follows:

This document describes an information model for Kerberos 5, but it does not directly describe any mapping onto a particular schema or modeling language. Hence, an implementation of this model consists of a mapping to such a language, e.g., an LDAP or SQL schema. Therefore, the standard normative key words require precise definition.

The terms "MUST" and "REQUIRED" mean that a schema implementing this model must have a way to represent a feature (i.e., that it is mandatory to implement it in the schema), 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 has to provide the feature and associated behavior consistent with the schema.

For instance, principalName (see Section 4.1.1.1) represents the name of 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" and "OPTIONAL" mean that it is optional for a KDC or administrative interface implementing this information model to implement this feature. These terms also mean that implementing the feature in the schema is optional.

Implementers of the schema should be aware that, unless the schema definition can represent critical but optional elements, language confusion may arise when optional elements are used but not understood by all implementations in a particular deployment.

The expression "MUST NOT be OPTIONAL" means that it is mandatory that a feature be implemented ("MUST" per the definition in [RFC2119]) and additionally that it must not be marked as optional in the schema language. In particular, this expression means that the feature is both mandatory to implement and must be present in all representations of the object to which it applies.

The terms "SHOULD" and "RECOMMENDED" mean that the consequences of not implementing the feature as if it were described with the "MUST" keyword must be carefully weighed before choosing a different course. In particular, these terms imply that interoperability concerns may arise from not following the recommended practice in schema that implement this model.

Context will determine whether the "SHOULD" key word applies to the schema, to the underlying behavior of the KDC, or both. For instance, when this document states that principalIsDisabled (see Section 4.1.1.4) SHOULD default to FALSE, this statement implies both a recommendation for the behavior of KDCs as well as a recommendation for the representation of that behavior in schema.

3. Information Model Demarcation

The information model specified in Section 4 describes objects, their properties, and the relationships between the 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 the ordering specified between the 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 in 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. For example, 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 with 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 (see Section 2.1.1 of [RFC1964]), including, if applicable for the name type, the realm.

The attribute MAY be multivalued if the implementation supports aliases, enterprise names, or both. In this case, exactly one of the principalName values MAY be designated as the canonical principalName. If the implementation supports encryption types (enctypes) that require salt, exactly one of the values of principalName MAY be designated as the canonical salting principalName.

Implementations (i.e., schema) that support enterprise names, aliases, or both, SHOULD provide for efficient lookup of principal objects based on the alias or 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. principalLastCredentialChangeTime

This single-valued attribute contains the time of the last successful change of credentials (e.g., password or private key) associated with this principal. The syntax of the attribute MUST be Internet date/time format from [RFC3339].

4.1.1.6. 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].

4.1.1.7. principalModifyTime

This single-valued attribute contains the time and date when this principal was last modified, excluding changes to credentials. The syntax of the attribute MUST be Internet date/time format from [RFC3339].

4.1.1.8. principalMaximumTicketLifetime

This single-valued attribute contains the time, in seconds, representing the maximum lifetime of a ticket issued for this principal.

4.1.1.9. 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.

4.1.1.10. principalAllowedEnctype

This OPTIONAL multivalued 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 KeySets and MAY be associated with 0 or more Policies. The KeySet is represented as an object in this model, because it has attributes associated with it

(the key version number). In typical situations, the principal is associated with exactly one KeySet, but implementations MUST NOT assume this case. That is, an implementation of this standard MUST be able to handle the general case of multiple KeySets 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 an 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. A schema 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

The key version number. This is a single-valued attribute containing a non-negative integer. This number is incremented by one each time a key in the KeySet is changed.

4.2.2. KeySet: Associations

Each KeySet MUST be associated with a set of one or more Keys.

4.3. Key

Implementations of this model MUST NOT force keys to be represented. That is, a schema that REQUIRED a key to be present would not meet this constraint.

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 using the string-to-key method defined in Section 3 of [RFC3961].

4.3.1.5. keyNotUsedBefore

The key MUST NOT be used before this date. The syntax of the attribute MUST be semantically equivalent to the standard ISO date format ([RFC3339]). This attribute MUST be single-valued.

4.3.1.6. keyNotUsedAfter

The key MUST NOT be used after this date. The syntax of the attribute MUST be semantically equivalent to the standard ISO date format ([RFC3339]). This attribute MUST be single-valued.

4.3.1.7. keyIsDisabled

This is a boolean attribute that SHOULD be set to FALSE by default. If this attribute is TRUE, the key MUST NOT be used. The keyIsDisabled attributed 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 is REQUIRED of the implementation.

4.4. Policy

Implementations SHOULD implement policies, but MAY allow them to be OPTIONAL. The policy should be thought of as a "typed hole", i.e., as an opaque binary value paired with an identifier of the 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) [RFC4517]
- A URI [RFC3986]
- A UUID [RFC4122]

Implementations of this specification are expected to assign globally unique identifiers to the list of the standard policy below in accordance with best practices for identifier management for the schema language used.

4.4.1.2. policyIsCritical

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

4.4.1.3. policyContent

This optional single opaque binary value is 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 the policy types that are not directly representable in that implementation.

4.4.1.4. policyUse

This optional single enumerated string value is 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 provide 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.

4.4.2.2. Password Management Policy

Password management policy controls how passwords are changed.

4.4.2.3. Keying Policy

A keying policy specifies the association of enctypes with new principals. For example, when a principal is created, one of the applicable keying policies is used to determine the set of keys to associate with the principal.

4.4.2.4. Ticket Flag Policy

A ticket flag policy specifies the ticket flags allowed for tickets issued for a principal.

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 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 [PASSWORD].

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 that exposes all non-key objects as entries and attributes. As in the example above, all keys are modified using the set/change password protocol [PASSWORD]. In this scenario, the implementation would typically not use a traditional LDAP implementation, but would treat LDAP as a protocol to access 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 situation demonstrates the value of creating an abstract information model that is mappable to multiple schema representations.

5.4. NETCONF

Given a YAML (YAML Ain’t Markup Language) 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 be treated as write-only attributes that SHALL be modified (using the administrative interface) only by the change-password protocol [PASSWORD].

Exposing the object model of a KDC typically implies that objects can be modified, deleted, or both. In a KDC, not all principals are created equal. 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 situation implies only 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 that have elevated rights.

7. Acknowledgments

The author wishes to extend his thanks to Love Hoernquist-Aestrand and Sam Hartman for their important contributions to this document.

8. References

8.1. 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.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

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

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC4517] Legg, S., "Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules", RFC 4517, June 2006.

8.2. Informative References

[PASSWORD] Williams, N., "Kerberos Set/Change Key/Password Protocol Version 2", Work in Progress, November 2008.

[RFC4510] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

Author's Address

Leif Johansson
Swedish University Network
Thulegatan 11
Stockholm
Sweden

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