NETWORK WORKING GROUP
Internet-Draft
Expires: September 6, 2007

N. Williams
Sun

March 5, 2007

Kerberos Set/Change Key/Password Protocol Version 2
draft-ietf-krb-wg-kerberos-set-passwd-06.txt

Status of this Memo

Copyright Notice

Abstract

This document specifies an extensible protocol for setting keys and changing the passwords of Kerberos V principals.

Table of Contents

1. Conventions used in this document

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].

2. Introduction

Up to this point Kerberos V has lacked a single, standard protocol for changing passwords and keys. While several vendor-specific protocols exist for changing Kerberos passwords/keys, none are properly internationalized and all are incomplete in one respect or another and none are sufficiently extensible to cope with new features that may be added to Kerberos V at some future time.

This document defines a protocol that is somewhat backward-compatible with the "kpasswd" protocol defined in [RFC3244] that uses more or less the same protocol framing.

This new protocol is designed to be extensible and properly internationalized.

3. The Protocol

The structure of the protocol is quite similar to that of typical RPC protocols. Each transaction consists of a data structure specific to an operation which is then wrapped in a data structure which is general to all operations of the protocol. These data structures are defined with the Abstract Syntax Notation 1 (ASN.1) [CCITT.X680.2002] and they are encoded using the Distinguished Encoding Rules (DER) [CCITT.X690.2002].

All protocol data is wrapped KRB-PRIV messages, or, in some cases, a KRB-ERROR, and framed in a header that is backwards compatible with [RFC3244].

3.1. Transports

The service supports only connection-oriented transports, specifically TCP, and SHOULD accept requests on TCP port 464, the same as in [RFC3244].

3.1.1. Protocol Framing

Requests and responses are exchanged using the same framing as in [RFC3244], but with the following differences:

- the protocol number field MUST be set to 0x2 (not 0xff80 or 0x1)
- the 'AP-REQ length' field of the request can be set to 0x0, in which case the 'AP-REQ' field of the request is excluded
- the 'KRB-PRIV' field of the request and reply is mutually exclusive with the 'AP-REQ' field of the request
- the 'AP-REP length' field of the reply can be set to 0x0, in which case the 'AP-REP' field of the reply is excluded
- all errors MUST be sent in a KRB-PRIV if the client's AP-REQ can be or has been accepted by the server
- any KRB-ERROR messages are framed and sent in the 'AP-REP' field of the reply

The initial message from the client MUST carry an AP-REQ and the response to any request bearing an AP-REQ MUST carry an AP-REP or MUST be a KRB-ERROR.

Subsequent messages exchanged on the same TCP connection MAY involve Kerberos V AP exchanges, but generally the client SHOULD NOT initiate

a new AP exchange except when it desires to authenticate as a different principal, when the ticket last used for authentication expires or when the server responds with an error indicating that the client must re-authenticate.

3.2. Protocol Version Negotiation

There are several major versions of this protocol. Version 2 also introduces a notion of protocol minor versions for use in negotiating protocol extensions. As of this time only one minor version is defined for major version 2: minor version 0, defined herein.

3.2.1. Protocol Major Version Negotiation

Version 2 clients that also support other versions, such as 0xff80,

as in [RFC3244], SHOULD attempt to use version 2 of the protocol first.

Servers which do not support version 2 of this protocol typically include their preferred version number in the reply and/or may include a reply in the e-data of a KRB-ERROR, or in a KRB-PRIV with a status code of KRB5_KPASSWD_MALFORMED.

Note that some [RFC3244] server implementations close the TCP connection without returning any other response. Note also that there is no integrity protection for the major version number in the protocol framing or for any data in a KRB-ERROR.

As a result change password protocol major version negotiation is subject to downgrade attacks. Therefore major version negotiation is NOT RECOMMENDED.

Where the server indicates that it does not support version 2, the client MAY, but SHOULD NOT, unless configured to do so, fall back on another major version of this protocol.

Version 2 servers MAY respond to non-v2 requests using whatever response is appropriate for the versions used by the clients, but if a server does not do this or know how to do this then it MUST respond with an error framed as in Section 3.1.1, using an AP-REP and KRB-PRIV if the client's AP-REQ can be accepted, or a KRB-ERROR otherwise and using a ProtocolErrorCode value of unsupported-major-version.

It is expected that implementations of as yet unspecified future major versions of this protocol will be required to support version 2 integrity protected error replies for properly indicating no support for version 2 of the protocl. We also hope that no further major versions of this protocol will be needed.

3.2.2. Protocol Minor Version Negotiation

Version 2 clients are free to use whatever protocol minor version and message extensions are available to them in their initial messages to version 2 servers, provided that the minor versions (other than 0) have been defined through IETF documents.

Version 2 servers MUST answer with the highest protocol minor version

number supported by the server and the client.

Version 2 clients MUST use the protocol minor version used in a server's reply for any subsequent messages in the same TCP session.

See Section 3.6 for further description of the protocol's extensibility and its relation to protocol minor versions and the negotiation thereof.

3.3. Use of Kerberos V and Key Exchange

This protocol makes use of messages defined in [RFC4120]. Specifically, AP-REQ, AP-REP, KRB-ERROR and KRB-PRIV.

All operations are to be performed by the server on behalf of the client principal.

Clients SHOULD use "kadmin/setpw" as the principal name of the server for all requests except when changing the client principal's own expired password, for which they should use "kadmin/changepw". The "kadmin/changepw" service exists to allow KDCs to limit principals with expired passwords to getting initial tickets to the password changing service only and only for changing expired passwords.

Servers MUST limit clients that used the "kadmin/changepw" service principal name to changing the password of the client principal.

The client MUST request mutual authentication and the client MUST MUST request the use of sequence numbers.

Servers MAY force the use of INITIAL or fresh tickets for any requests -- see Section 4.3. Traditionally users with expired passwords are allowed only INITIAL tickets for the password changing server, therefore clients MUST support the use of INITIAL tickets.

Servers MUST specify a sub-session key.

The encrypted part of KRB-PRIVs MUST be encrypted with the server's sub-session key and key usage 20 (client->server) or 21 (server->client).

After each new AP exchange the client and server MUST destroy the

session keys, if any, resulting from the previous AP exchange.

3.4. Use of ASN.1

This protocol's messages are defined in ASN.1, using only features from [CCITT.X680.2002]. All ASN.1 types defined herein are to be encoded in DER [CCITT.X690.2002]. A complete ASN.1 module is given in Section 5.

The DER encoding of the ASN.1 PDUs are exchanged wrapped in a KRB-PRIV as described above and/or as e-data in KRB-ERROR messages.

3.5. Internationalization

This protocol's request PDU carries an optional field indicating the languages spoken by the client user; the client SHOULD send its list of spoken languages to the server (once per-TCP session).

The server SHOULD localize all strings intended for display to users to a language in common with the languages spoken by the client user.

Strings for Kerberos principal and realm names used in this protocol are be constrained as per [RFC4120].

3.5.1. Normalization Forms for UTF-8 Strings

Because Kerberos V [RFC4120] restricts principal names, realm names and passwords to IA5String, this protocol uses UTF8String with an extensible constraint to IA5String.

Future versions of Kerberos may relax this constraint; if so then a minor version of this protocol should relax this constraint accordingly.

3.5.2. Language Negotiation

The server MUST pick a language from the client's input list or the default language tag (see [RFC3066]) for text in its responses which is meant for the user to read.

The server SHOULD use a language selection algorithm such that consideration is first given to exact matches between the client's spoken languages and the server's available locales, followed by "fuzzy" matches where only the first sub-tags of the client's language tag list are used for matching against the servers available locales.

Servers MUST cache the optional language tag lists from prior requests for use with subsequent requests that exclude the language tag list. Clients MAY expect such server behaviour and send the language tag lists only once per-TCP session. Clients SHOULD send the server the language tag list at least once.

When the server has a message catalog for one of the client's spoken languages the server SHOULD localize any text strings intended for display to users.

3.6. Protocol Extensibility

The protocol is defined in ASN.1 and uses extensibility markers throughout. As such, the module presented herein can be extended within the framework of [CCITT.X680.2002].

Typed holes are not used in this protocol as it is very simple and does not require the ability to deal with abstract data types defined in different layers. For this reason, the only way to extend this protocol is by extending the ASN.1 module within the framework of the IETF; all future extensions to this protocol have to be defined in IETF documents unless otherwise specified in a future IETF revision of this protocol.

A protocol minor version number is used to negotiate use of extensions. See Section 3.2.2 for the minor version negotiation.

Servers SHOULD ignore unknown additions to the ASN.1 types, in initial requests, where the syntax allows them, except for extensions to the "Op-req" type, which MUST result in an error.

Servers MUST respond with an error (ProtocolErrorCode value of proto-unsupported-operation) to clients that use operations unknown to the server.

3.7. Protocol Subsets

The structure of the protocol is such that the ASN.1 syntaxes for the various operations supported by the protocol are independent of the each other. Client and server implementations MAY implement subsets of the overall protocol by removing some alternatives to the Op-req, Op-rep and Op-err CHOICEs from the ASN.1 module given in Section 5.

For example, it should be possible to have a password-change only client that cannot set principal's keys - and vice versa.

4. Protocol Elements

The protocol as defined herein supports the following operations relating to the management of Kerberos principal's passwords or keys:

- change password (or enctypes and string-to-key parameters)
- set password (administrative)
- set new keys
- generate new keys
- get new, un-committed keys
- commit new keys
- get password policy name and/or description of principal
- list aliases of a principal
- list enctypes and version of Kerberos V supported by realm

The operation for retrieving a list of aliases of a principal is needed where KDCs implement aliasing of principal names and allows clients to properly setup their key databases when principal aliasing is in use.

Operations such as creation or deletion of principals are outside the scope of this document, and should be performed via other means, such as through directories or other Kerberos administration protocols.

The individual operations are described in Section 4.3.

4.1. Password Quality Policies

Servers may reject new user passwords for failing to meet password quality policies. When this happens the server must be able to communicate the policy to the user so that the user may select a

better password.

The protocol allows for two ways to do this:

- through error codes that identify specific password quality policies known;
- through localized error strings.

The use of localized error strings allows servers to convey non-standard password quality policies to users, but it requires server-side message catalogs for localization and support for language negotiation. The use of error codes only allows for standard password quality policies that clients must be aware of a priori, therefore use of error codes requires the distribution of new message catalogs to clients whenever new error codes are added; this simplifies servers at the expense of password quality extensibility.

When a server would reject a password due to its failing to meet a standard password quality policy the the server MUST use the appropriate error codes (see below) and the server SHOULD send a localized error message to the user.

When a server would reject a password due to its failing to meet a non-standard password quality policy (one not described herein) the the server MUST send a localized error message to the user.

4.1.1. Standard Password Quality Policies

- pwq-too-soon

  It is too soon for the principal to change its password or long-term keys.

- pwq-history

  The new password is one that the principal has used before or is too similar to a password that it has used before.

- pwq-too-short

  The new password is too short.

- pwq-dictionary-words

  The new password is or contains words that are in the dictionary.

- pwq-prohibited-codepoints

  The new password contains prohibited codepoints.

- pwq-need-more-char-classes

  The new password does not have characters from enough character classes (lower-case letters, upper-case letters, digits, punctuation, etc...).

4.2. PDUs

The types "Request," "Response" and "Error-Response" are the ASN.1 module's PDUs.

The "Request" and "Response" PDUs are always to be sent wrapped in KRB-PRIV messages, except for the "Error-Response" PDU which MUST be sent as KRB-ERROR e-data (see Section 3.3) when AP exchanges fail, otherwise it MUST be sent wrapped in a KRB-PRIV.

The ASN.1 syntax for the PDUs is given in Section 5.

Note that the first field of each PDU is the major version of the protocol, defaulted to 2, meaning that it is never included in version 2 exchanges. Similarly, the second field of each PDU is the minor version, defaulted to 0.

The request, responses and error PDUs consist of an outer structure

("Request," "Response" and "Error-Response") containing fields common to all requests, responses and errors, respectively, and an inner structure for fields that are specific to each operation's requests/responses. The inner structure is optional in the case of the Error-Response PDU and need not be included when generic errors occur for which there is a suitable ProtocolErrorCode.

Specifically, the outer Request structure has a field for passing a client user's spoken (read) languages to the server. It also has two optional fields for identifying the requested operation's target principal's name and realm (if not sent then the server MUST use the client's principal name and realm as the target). A boolean field for indicating whether or not the request should be dry-run is also included; dry-runs can be used to test server policies, and servers MUST NOT modify any principals when processing dry-run requests.

The Response and Error PDUs' outer structures include a field indicating the language that the server has chosen for localization of text intended to be displayed to users; this field is defaulted to "i-default". This language tag applies to all UTF8 strings in the inner structure (Op-rep and Op-err) that are meant to be displayed to users.

The protocol error codes are:

- proto-generic-error

  An operation-specific error ocurred, see the inner Op-error.

- proto-format-error

  The server failed to parse a message sent by the client.

- proto-unsupported-major-version

  The server does not support the major version of this protocol requested by the client.

- proto-unsupported-minor-version

  The server does not support the minor version of this protocol

requested by the client.

- proto-unsupported-operation

  The server does not support the operation requested by the client.

- proto-wrong-service-principal

  Use kadmin/setpw for the server's principal name.

- proto-re-authentication-required

  The server demands that the client re-authenticate through a new AP exchange.

- proto-initial-ticket-required

  Use of an INITIAL ticket is required for the requested operation.

- proto-client-and-target-realm-mismatch

  The server requires that the client's principal name and the target principal of the operation share the same realm name.

- proto-target-principal-unknown

  The target of the client's operation is not a valid principal.

- proto-authorization-failed

  The client principal is not authorized to perform the requested operation.

- proto-fresh-ticket-required

  The server would like the client to re-authenticate using a fresh ticket.

4.3. Operations

This section describes the semantics of each operation request and

response defined in the ASN.1 module in Section 5.

4.3.1. Null

4.3.2. Change Kerberos Password

4.3.3. Set Kerberos Password

4.3.4. Set Kerberos Keys

4.3.5. Generate Kerberos Keys

4.3.6. Get New Keys

4.3.7. Commit New Keys

4.3.8. Get Password Quality Policy

4.3.9. Get Principal Aliases

4.3.10. Get Realm's Supported Kerberos V Version and Features

4.3.11. Retrieve Principal's S2K Params and Preferred Params

4.4. Principal Aliases

Applications that use Kerberos often have to derive acceptor principal names from hostnames entered by users. Such hostnames may be aliases, they may be fully qualified, partially qualified or not qualified at all. Some implementations have resorted to deriving principal names from such hostnames by utilizing the names services to canonicalize the hostname first; such practices are not secure unless the name service are secure, which often aren't.

One method for securely deriving principal names from hostnames is to alias principals at the KDC such that the KDC will issue tickets for principal names which are aliases of others. It is helpful for principals to know what are their aliases as known by the KDCs.

Note that changing a principal's aliases is out of scope for this protocol.

4.5. Kerberos V Feature Negotiation

Principals and realms' KDCs may need to know about additional Kerberos V features and extensions that they each support. Several operations (see above) provide a way for clients and servers to exchange such infomration, in the form of lists of types supported for the various typed holes used in Kerberos V.

5. ASN.1 Module

6. Security Considerations

Implementors and site administrators should note that the redundancy of UTF-8 encodings of passwords varies by language. Password quality policies SHOULD, therefore, take this into account when estimating the amount of redundancy and entropy in a proposed new password.

Kerberos set/change password/key protocol major version negotiation cannot be done securely; a downgrade attack is possible against clients that attempt to negotiate the protocol major version to use with a server. It is not clear at this time that the attacker would gain much from such a downgrade attack other than denial of service (DoS) by forcing the client to use a protocol version which does not support some feature needed by the client (Kerberos V in general is subject to a variety of DoS attacks anyways [RFC4120]). Clients SHOULD NOT negotiate support for legacy major versions of this protocol unless configured otherwise.

This protocol does not have Perfect Forward Security (PFS). As a result, any passive network snooper watching password/key changing operations who has stolen a principal's password or long-term keys can find out what the new ones are.

7. References

7.1. Normative References

[CCITT.X680.2002]
International International Telephone and Telegraph Consultative Committee, "Abstract Syntax Notation One (ASN.1): Specification of basic notation", CCITT Recommendation X.680, July 2002.

[CCITT.X690.2002]
International International Telephone and Telegraph Consultative Committee, "ASN.1 encoding rules: Specification of basic encoding Rules (BER), Canonical encoding rules (CER) and Distinguished encoding rules (DER)", CCITT Recommendation X.690, July 2002.

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

[RFC3066] Alvestrand, H., "Tags for the Identification of Languages", RFC 3066, January 2001.

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

7.2. Informative References

[RFC3244] Swift, M., Trostle, J., and J. Brezak, "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols", RFC 3244, February 2002.

Author's Address

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US

Email: Nicolas.Williams@sun.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).