

Kerberos Change Password Protocol

Status of this Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``[id-abstracts.txt](#)'' listing contained in the Internet-Drafts Shadow Directories on [ds.internic.net](#) (US East Coast), [nic.nordu.net](#) (Europe), [ftp.isi.edu](#) (US West Coast), or [munnari.oz.au](#) (Pacific Rim).

Distribution of this memo is unlimited. Please send comments to the [<cat-ietf@mit.edu>](#) mailing list.

Abstract

The Kerberos V5 protocol [[RFC1510](#)] does not describe any mechanism for users to change their own passwords. In order to promote interoperability between workstations, personal computers, terminal servers, routers, and KDC's from multiple vendors, a common password changing protocol is required.

Overview

When a user wishes to change his own password, or is required to by local policy, a simple request of a password changing service is necessary. This service must be implemented on at least one host for each Kerberos realm, probably on one of the kdc's for that realm. The service must accept requests on UDP port 464 (kpasswd), and may accept requests on TCP port 464 as well.

The protocol itself consists of a single request message followed by a single reply message. For UDP transport, each message must be fully contained in a single UDP packet.

Request Message



message length (16 bits)

Contains the length of the message, including this field, in bytes (big-endian integer)

protocol version number (16 bits)

Contains the hex constant 0x0001 (big-endian integer)

AP-REQ length (16 bits)

length (big-endian integer) of AP-REQ data, in bytes.

AP-REQ data, as described in [RFC1510](#) (variable length)

This AP-REQ must be for the service principal `kadmin/changepw@REALM`, where REALM is the REALM of the user who wishes to change his password. The Ticket in the AP-REQ must be derived from an AS request (thus having the INITIAL flag set), and must include a subkey in the Authenticator.

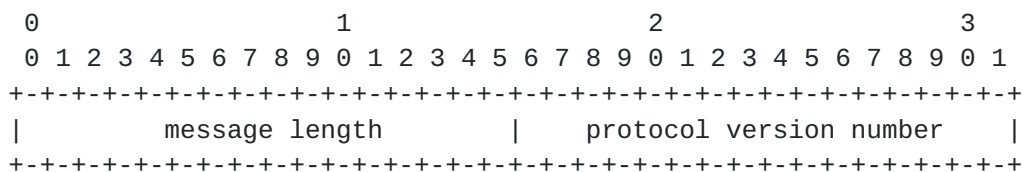
KRB-PRIV message, as described in [RFC1510](#) (variable length)

This KRB-PRIV message must be generated using the subkey in the Authenticator in the AP-REQ data. The user-data component of the message must consist of the user's new password.

The server must verify the AP-REQ message, decrypt the new password, perform any local policy checks (such as password quality, history, authorization, etc.) required, then set the password to the new value specified.

The principal whose password is to be changed is the principal which authenticated to the password changing service. This protocol does not address administrators who want to change passwords of principal besides their own.

Reply Message



message length (16 bits)

This field should contain information which the server thinks might be useful to the user. No particular format is mandated,

and the field may be omitted if the server does not wish to include it. This field is analogous to the string which follows the numeric code in SMTP, FTP, and similar protocols.

Security Considerations

This document deals with changing passwords for Kerberos. Because Kerberos is used for authentication and key distribution, it is important that this protocol use the highest level of security services available to a particular installation. Mutual authentication is performed, so that the server knows the request is valid, and the client knows that the request has been received and processed by the server.

There are also security issues relating to dropped, stolen, or modified messages. An attacker (or simply a lossy network) could cause either the request or reply to be dropped, or substitute a KRB-ERROR message in the reply.

If a reply is dropped, it is reasonable for the client to construct a new authenticator, re-encrypt the request, and retransmit. If the request was lost, the server will treat this as a valid request, and all should work normally. If the reply was lost, then the server should take care to notice that the request was a duplicate of the prior request, and indicate success without actually changing the password or any other information (such as modification timestamps).

If a success reply was replaced with an error reply, then one would expect that the user would attempt the operation again. Again, the server should recognize the request as a duplicate and indicate success without changing the password. If the user is required to provide the old password again, but the password was actually changed successfully, then some user confusion could result. This is, unfortunately, impossible to prevent.

References

[RFC1510] Kohl, J. and Neuman, C., "The Kerberos Network Authentication Service (V5)", [RFC 1510](#), September 1993.

Author's Address

Marc Horowitz
Stonecast, Inc.
108 Stow Road
Harvard, MA 01451

Phone: +1 978 456 9103
Email: marc@stonecast.net

