INTERNET-DRAFT
draft-ietf-cat-kerberos-set-passwd-00.txt
expires March, 2000

Mike Swift
Microsoft
Jonathan Trostle
Cisco Systems
John Brezak
Microsoft

Extending Change Password for Setting Kerberos Passwords

0. Status Of This Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

Comments and suggestions on this document are encouraged. Comments on this document should be sent to the CAT working group discussion list:

ietf-cat-wg@stanford.edu

This document expires in March, 2000.

1. Abstract

The Kerberos [1] change password protocol [2], does not allow for an administrator to set a password for a new user. This functionality is useful in some environments, and this proposal extends [2] to allow password setting. The changes are: adding new fields to the request message to indicate the principal which is having its password set, not requiring the initial flag in the service ticket, using a new protocol version number, and adding three new result codes.

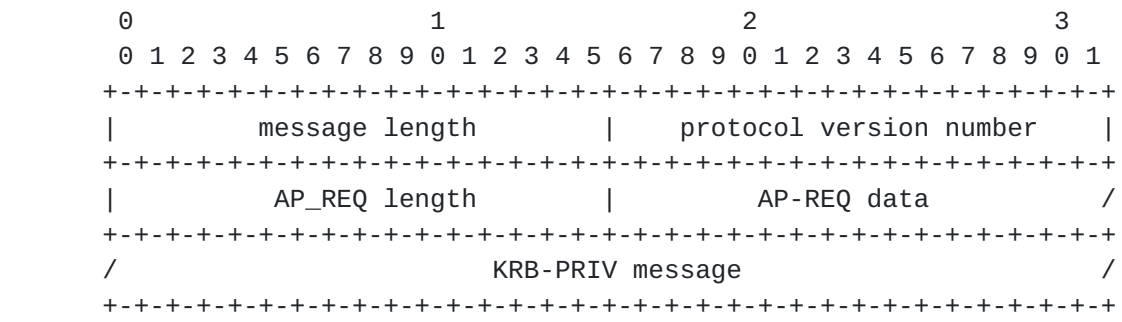
2. The Protocol

The service must accept requests on UDP port 464 and TCP port 464 as

well. The protocol 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.

For TCP transport, there is a 4 octet header in network byte order precedes the message and specifies the length of the message. This requirement is consistent with the TCP transport header in 1510bis.

Request Message



All 16 bit fields are in big-endian order.

message length field: contains the number of bytes in the message including this field.

protocol version number: contains the hex constant 0xff80 (big-endian integer).

AP-REQ length: length of AP-REP data, in bytes. If the length is zero, then the last field contains a KRB-ERROR message instead of a KRB-PRIV message.

AP-REQ data: (see [1]) The AP-REQ message must be for the service principal kadmin/changepw@REALM, where REALM is the REALM of the user who wishes to change/set his password. The ticket in the AP-REQ must must include a subkey in the Authenticator.

KRB-PRIV message (see [1]) This KRB-PRIV message must be generated using the subkey from the authenticator in the AP-REQ data. The initial flag of the service ticket is ignored by the server unless policy dictates otherwise, in which case the request will be rejected with result code 0x0007 if the initial flag is not set and is required for this particular request. The user-data component of the message consists of the following ASN.1 structure encoded as an OCTET STRING:

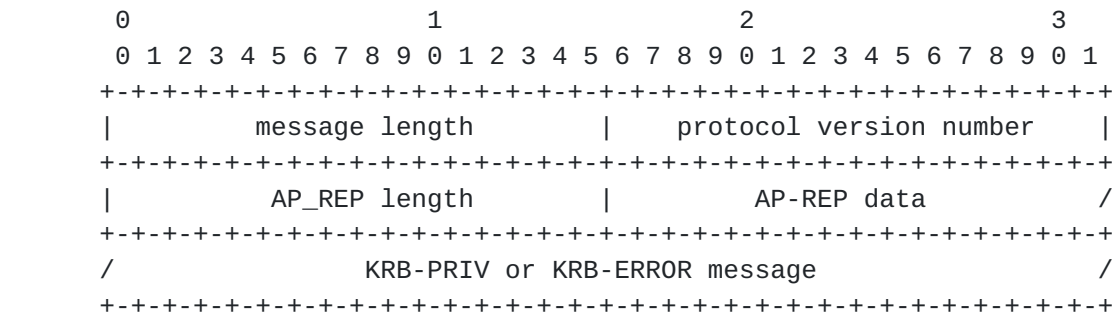
```
ChangePasswdData ::=  SEQUENCE {
                      newpasswd[0]   OCTET STRING,
                      targname[2]    PrincipalName OPTIONAL,
                      targrealm[3]   Realm OPTIONAL
                      }
```

The server must verify the AP-REQ message, check whether the client

principal in the ticket is authorized to set/change the password (either for that principal, or for the principal in the targname field if present), and decrypt the new password. The server also checks whether the initial flag is required for this request, replying with status 0x0007 if it is not set and should be. An authorization failure is cause to respond with status 0x0005. For forward compatibility, the server should be prepared to ignore fields after targrealm in the structure that it does not understand.

The newpasswd field contains the cleartext password, and the server should apply any local policy checks including password policy checks. The server then generates the appropriate keytypes from the password and stores them in the KDC database. If all goes well, status 0x0000 is returned to the client in the reply message (see below).

Reply Message



All 16 bit fields are in big-endian order.

message length field: contains the number of bytes in the message including this field.

protocol version number: contains the hex constant 0x0001 (big-endian integer). (The reply message has the same format as in [2]).

AP-REP length: length of AP-REP data, in bytes. If the length is zero, then the last field contains a KRB-ERROR message instead of a KRB-PRIV message.

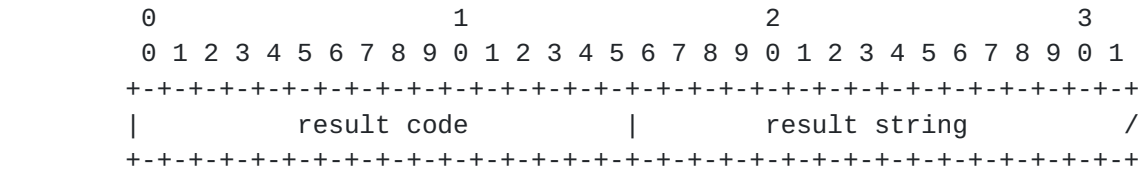
AP-REP data: the AP-REP is the response to the AP-REQ in the request packet.

KRB-PRIV or KRB-ERROR message: - from [2]: if the AP-REP length is zero, then this field contains a KRB-ERROR message. Otherwise, it contains a KRB-PRIV message. This KRB-PRIV message must be generated using the subkey in the authenticator in the AP-REQ data.

The server will respond with a KRB-PRIV message unless it cannot decode the client AP-REQ or KRB-PRIV message, in which case it will respond with a KRB-ERROR message.

The user-data component of the KRB-PRIV message, or e-data component

of the KRB-ERROR message, must consist of the following data.



result code (16 bits) (result codes 0-4 are from [2]):
The result code must have one of the following values (big-endian integer):

KRB5_KPASSWD_SUCCESS	0	request succeeds (This value is not allowed in a KRB-ERROR message)
KRB5_KPASSWD_MALFORMED	1	request fails due to being malformed
KRB5_KPASSWD_HARDERROR	2	request fails due to "hard" error in processing the request (for example, there is a resource or other problem causing the request to fail)
KRB5_KPASSWD_AUTHERROR	3	request fails due to an error in authentication processing
KRB5_KPASSWD_SOFTERROR	4	request fails due to a "soft" error in processing the request
KRB5_KPASSWD_ACCESSDENIED	5	requestor not authorized
KRB5_KPASSWD_BAD_VERSION	6	protocol version unsupported
KRB5_KPASSWD_INITIAL_FLAG_NEEDED	7	initial flag required

0xFFFF if the request fails for some other reason.
Although only a few non-zero result codes are specified here, the client should accept any non-zero result code as indicating failure.

result string - from [2]:
This field should contain information which the server thinks might be useful to the user, such as feedback about policy failures. The string must be encoded in UTF-8. It may be omitted if the server does not wish to include it. If it is present, the client should display the string to the user. This field is analogous to the string which follows the numeric code in SMTP, FTP, and similar protocols.

3. Bibliography

[1] J. Kohl, C. Neuman. The Kerberos Network Authentication Service (V5). Request for Comments 1510.

[2] M. Horowitz. Kerberos Change Password Protocol. ftp://ds.internic.net/internet-drafts/draft-ietf-cat-kerb-chg-password-02.txt

4. Expiration Date

This draft expires on March 31, 2000.

5. Authors' Addresses

Jonathan Trostle
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134

Email: jtrostle@cisco.com, jtrostle@world.std.com

Mike Swift
1 Microsoft Way
Redmond, WA 98052
mikesw@microsoft.com

John Brezak
1 Microsoft Way
Redmond, WA 98052
jbrezak@microsoft.com