Kerberos Working Group
Internet Draft
Document: draft-swift-win2k-krb-user2user-01.txt
Category: Informational

M. Swift
University of WA
J. Brezak
Microsoft
P. Moore
Sandia National Labs

March 2001

User to User Kerberos Authentication using GSS-API

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

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.

1. Abstract

This draft documents a simple extension to the Kerberos GSS-API mechanism to support user to user authentication both in the case where the client application explicitly requests user to user authentication and when it does not know whether the server supports user to user authentication.

2. 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 RFC-2119 [2].

3. Introduction

The Kerberos user to user authentication mechanism allows for a client application to connect to a service that is not in possession of a long term secret key. Instead, the session ticket from the KERB-AP-REQ is encrypted using the session key from the service's ticket granting ticket. According to RFC 1510 [3]:

> If the ENC-TKT-IN-SKEY option has been specified and an additional ticket has been included in the request, the KDC will decrypt the additional ticket using the key for the server to which the additional ticket was issued and verify that it is a ticket-granting ticket. If the request succeeds, the session key from the additional ticket will be used to encrypt the new ticket that is issued instead of using the key of the server for which the new ticket will be used (This allows easy implementation of user-to-user authentication, which uses ticket-granting ticket session keys in lieu of secret server keys in situations where such secret keys could be easily compromised).

RFC2078 [5], in section 5.2, discusses a ôDouble-TGT K-5ö mechanism and scenario, but not in the detail required in order to implement the mechanism. The RFC for the Kerberos V5 GSS-API mechanism at the time this draft was prepared, RFC 1964 [4] does not support user-to-user authentication.

This draft provides details as to mechanism type, token identifiers, messages and message types sufficient to document an implementation of user-to-user authentication in Kerberos GSS-API. It follows the scenario described in RFC2078.

The approach documented in this draft has been used to support user-to-user authentication in the Microsoft Windows 2000 SSPI with the Kerberos V5 protocol, and in a patched Kerberos V5 implementation being used to support a computing grid at Sandia, Lawrence Livermore, and Los Alamos National Laboratories.

4. User to User as a New Mechanism

A new mechanism OID may be used to establish a user-to-user session:

```
{iso(1) member-body(2) United States(840) mit(113554)
infosys(1) gssapi(2) krb5(2) usertouser(3)}
```

In the case that the client application knows that the server requires user-to-user authentication, then the initial call to GSS_Init_Sec_Context will request this mechanism. This new mechanism is used with a token exchange that extends the conventional Kerberos GSS-API protocol by adding an additional round trip to request the TGT from the service. As with all Kerberos GSS-API messages, the following tokens are encapsulated in the GSS-API framing. The first token of the exchange will have an innerContextToken with a 2-octet TOK_ID field containing 04 00 (KERB-TGT-REQUEST) followed by a Kerberos V5 message as follows:

```
KERB-TGT-REQUEST ::= SEQUENCE {
        pvno[0]                         INTEGER,
        msg-type[1]                     INTEGER,
        server-name[2]                  PrincipalName OPTIONAL,
        realm[3]                        Realm OPTIONAL
```

```
}
```

The TGT request consists of four fields:

pvno and msg-type are as defined in RFC1510 section 5.4.1. msg-type is KRB_TGT_REQ (16).

server-name : this field optionally contains the name of the server. If the client application doesn't know the server name this can be left blank and the server application will pick the appropriate server credentials which may be the default credential.

realm : this field optionally contains the realm of the server. If the client application doesn't know the server realm this field can be left blank and the server application will pick the appropriate server credentials which may be the server's default realm.

The server name and realm are included to allow a server application to act for multiple principles in different realms and to choose which credentials to use.

The response to the KERB-TGT-REQUEST message will be a KERB_TGT_REPLY token which will have an innerContextToken with a 2-octet TOK_ID field containing 04 01 (KERB-TGT-REPLY) followed by a Kerberos V5 message as follows:

```
     KERB-TGT-REPLY ::= SEQUENCE {
             pvno[0]                         INTEGER,
             msg-type[1]                     INTEGER,
             ticket[2]                       Ticket
             }
```

The TGT reply contains the following fields:

pvno and msg-type are as defined in RFC1510 section 5.4.1. msg-type is KRB_TGT_REP (17)

ticket : contains the TGT for the service specified by the server name and realm passed by the client or the default service.

If the service does not possess a ticket granting ticket, it should return the error KRB_AP_ERR_NO_TGT (0x43).

If the server name and realm in the KERB-TGT-REQUEST message do not match the name of the service, then the service should return the error KRB_AP_ERR_NOT_US.

Following the exchange of the TGT request messages, the initiator requests a ticket to the service from the KDC using a KERB-TGS-REQ with the KDCoption ENC-TKT-IN_SKEY and the second ticket in the

additional-tickets of the KDC-REQ-BODY. Upon receipt of the TGS-REP the rest of the authentication identical to the Kerberos GSS-API mechanism defined in RFC 1964 [4].

5. User-to-User when applied via KDC policy

Implementations MAY support the ability apply a policy on a user account such that the KDC will not allow conventional service ticket requests, and when presented with a KERB_TGS_REQ that does not contain a second ticket with an ENC_TKT_IN_SKEY KDCoption will respond with a KRB-ERROR with the msg-type KDC_ERR_MUST_USE_USER2USER (or KRB5PLACEHOLD_27).

In this case, the client need not explicitly request user-to-user in order to get a user-to-user connection. Implementations may use this error code to set a flag and return a GSS_C_CONTINUE_NEEDED so that the next round uses the mechanism described in section 4.

6. User to User when applied by server policy

In the case that the client application doesn't know that a service requires user-to-user authentication, and requests and receives a conventional KRB_AP_REP, the client will send the KRB_AP_REP request, and the server will respond with a KRB_ERROR token as described in RFC1964, with a msg-type of KRB_AP_ERR_USER_TO_USER_REQUIRED (0x45). The server may optionally pass the TGT in the data field of this error message. In response to this error, the initiator sets flags and returns a GSS_C_CONTINUE_NEEDED so that the next round uses the mechanism described in section 4.

7. Security Considerations

These extensions simply enable an existing Kerberos 5 authentication protocol so that it may be used from GSSAPI.

There is some risk in a server handing out its ticket-granting-ticket to any client that requests it, in that it gives an attacker a piece of encrypted material to decrypt. However, the same material may be obtained from listening to any legitimate client connection.

It should be noted here that the exchange described in section 6 requires that the KDC provide tickets for user accounts, which will contain known plaintext encrypted in the usersÆ private key. The risk associated with this is entirely mitigated where a KDC supports the KDC_MUST_USE_USER2USER feature, which allows a restriction on user accounts to ensure that all tickets for that account are encrypted in the TGT session key, and not the long term key of the user.

8. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

3 J. Kohl, C. Neuman, "The Kerberos Network Authentication Service(V5)", RFC 1510.

4 J. Linn, "The Kerberos Version 5 GSS-API Mechanism", RFC 1964

5 J. Linn, ôGeneric Security Service Application Program Interface, Version 2ö, RFC 2078

9. Author's Addresses

Michael Swift
University of Washington
Seattle, Washington
Email: mikesw@cs.washington.edu

John Brezak
Microsoft
One Microsoft Way
Redmond, Washington
Email: jbrezak@microsoft.com

Patrick Moore
Sandia National Laboratories
PO Box 5800 Mail Stop
Albuquerque, New Mexico
Email: pcmoore@sandia.gov

Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."