CAT Working Group
INTERNET-DRAFT
<draft-ietf-cat-iakerb-00.txt >
Expires April 30, 1998

Michael M. Swift
Microsoft

October, 31, 1997

Initial Authentication with Kerberos and the GSS-API (IAKERB)

STATUS OF THIS MEMO

ABSTRACT

This draft proposes a new Kerberos authentication mechanism for use when the client computer is unable to contact a Key Distribution Center (KDC). Instead, the client will send Authentication Service (AS) and Ticket Granting Service (TGS) requests to the server, which will then forward them to the appropriate KDC.

Table of Contents

1. Introduction

The standard Kerberos mechanism works well in a LAN environment where clients are well connected and can quickly locate and communicate with network services such as the KDC. Unlike many other authentication protocols, Kerberos requires that the client do most of the work of authentication by locating and calling a KDC to obtain tickets. All a server must do is to decrypt the AP request and verify that it is not a replay

However, in certain circumstances this is not a good use of computer resources. On the Internet, for example, servers tend to be far better connected and more able to locate a KDC then clients are. Similarly, when dialing up to an Internet Service Provider (ISP) the client computer is essentially unconnected while the ISP's computer are well connected to the Internet as well as other servers locally. Hence, it makes sense in these situations to allow the client to forward KDC requests to the server and let the server communicate with the KDC.

2. Basic Protocol

The mechanism ID for user to user GSS-API Kerberos, in accordance with the mechanism proposed by SPNEGO for negotiating protocol variations, is:

```
 {iso(1) member-body(2) United States(840) mit(113554)
infosys(1) gssapi(2) krb5(2) initialauth(4)}
```

The basic protocol is the existing exchanges between clients and the KDC detailed in RFC1510 [1]. The first context message is an AS request, to which the server responds with an AS reply. The client may either request a TGT during the AS request or directly request a session ticket if the connection is for a short period, only one service will be contacted, and the service principal and client principal are both in the same realm. Otherwise, the client will use the TGT it initially obtained and use it to create further TGS requests which will also be sent to the server as context messages.

As with all Kerberos GSS-API messages, the following tokens are encapsulated in the GSS-API framing. In addition, the innerContextToken field of the context establishment tokens contain the context message preceded by a 2-byte TOK_ID field. The messages and their respective IDs are listed below.

Message	TOK_ID
KRB-AS-REQ	05 00
KRB-AS-REP	05 01
KRB-TGS-REQ	05 02
KRB-TGS-REP	05 03

3.   Addresses in Tickets

In IAKERB, the machine sending requests to the KDC is the server and not the client. As a result, the client should not include its addresses in any KDC requests for two reasons. First, the, the KDC may reject the forwarded request as being from the wrong client. Second, in the case of initial authentication for a dial-up client, the client machine may not yet possess a network address. Hence, as allowed by RFC1510 [1], the addresses field of the AS and TGS requests should be blank and the caddr field of the ticket should similarly be left blank.

4.   Generating Initial Credentials

As this flavor of authentication uses AS requests, the client name, realm, and password must be available to the mechanism implementation. The GSS-API does not support passing in credentials to the GSS_acquire_cred_handle, and credentials are by their nature extemely package specific. Hence, it is left to the implementation to add an interface for setting the initial credentials.

5. Sample Usage Scenarios

Below are detailed three different scenarios using IAKERB and the messages sent in each case. In the first two cases the client never procures a ticket granting ticket. This is useful for an environment where communication is slow and the TGT would not later be used. In the third scenario the client procures a TGT first and uses it to request a ticket to the service. It is up to the implementation which variety to implement.

5.1 Case 1: Client and Server are in same realm

In this case, the first call to gss_init_sec_context() on the client generates an AS request with the client name set to the client's principal name and the server name set to the server's principal name. The client application sends this to the server application, which then calls gss_accept_sec_context(). The GSS runtime on the server forwards the request to the KDC, which responds with an AS reply. The runtime returns the AS reply from gss_accept_sec_context() and the service returns it to the client application.

The client application passes the AS reply to gss_init_sec_context(), which creates an AP request and packages it up identically to the format in RFC 1964 [2]. The client application then sends the AP request to the server, which calls gss_accept_sec_context() to verify the AP request.

```
Client                         Server          KDC

AS-REQ(cname,sname,realm)-->   forwards -->
                           <-- forwards    <--   AS-REP

AP-REQ -->                     Verifies AP request
```

5.2 Case 2: Client and Server in different realm

In this case, the client GSS runtime analyzes the target name and determines that it is from a different realm than the client. It then generates an AS request for a cross-realm TGT for the server's realm. The server runtime forwards the request to the client's KDC (C.KDC) and returns the AS reply containing a TGT for the server's realm. The client runtime then generates a TGS request for a ticket to the server with the cross-realm TGT. The server runtime forwards this to the server's KDC (S.KDC), which returns a session ticket to the server. The client runtime then generates a normal AP request for the server using this ticket.

```
Client                        Server          S.KDC   C.KDC

AS-REQ(cname,krbtgt/srealm,crealm)
                              forwards --------------->
                          <-- forwards        <------- AS-REP

TGS-REQ(krbtgt/srealm,server) forwards ---->
                          <-- forwards    <-- TGS-REP

AP-REQ -->                    Verifies AP request
```

5.3 Case 3: Client and Server in different realms with a TGT

In this case, the client plans on contacting additional services after authenticating with the server so it wants to obtain a TGT. The transaction is very similar to the previous example, but in this case the client obtains a TGT in its own realm before obtaining a cross-realm TGT for the server's realm.

```
Client                        Server          S.KDC   C.KDC

AS-REQ(cname,krbtgt/crealm,crealm)
      -->                     forwards --------------->
                          <-- forwards        <------- AS-REP

TGS-REQ(krbtgt/crealm,krbtgt\srealm)
      -->                     forwards --------------->
                          <-- forwards        <------- TGS-REP

TGS-REQ(krbtgt/srealm,server)
      -->                     forwards ---->
                          <-- forwards    <-- TGS-REP
```

```
AP-REQ -->                     Verifies AP request
```

6. Combining IAKERB with other Kerberos Extensions

This protocol is usable with other proposed Kerberos extensions such as PKINIT (Public Key Cryptography for Initial Authentication in Kerberos [3]) or User-to-User Kerberos [4]. In both cases, the messages which would normally be sent to the KDC by the GSS runtime are instead sent by the client application to the server, which then forwards them to a KDC.

7. Security Considerations

This variation on the Kerberos protocol does not change its security characteristics much. The biggest difference is the lack of addresses in the tickets. As addresses cannot be relied on to provide security but are at best make it more difficult to break a protocol, this is not a serious threat.

8. References

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

[2] J. Linn. The Kerberos Version 5 GSS-API Mechanism. Request for Comments 1964

[3] B. Tung, C. Neuman, J. Wray, A. Medvinsky, M. Hur, J. Trostle, Public Key Cryptography for Initial Authentication in Kerberos, draft-ietf-cat-kerberos-pk-init-04.txt.

[4] M. Swift, User to User Kerberos Authentication using GSS-API, draft-ietf-cat-user2user-01.txt.

Author's address

Michael Swift
Microsoft

1 Microsoft Way
Redmond, Washington, 98052, U.S.A.

Email: mikesw@microsoft.com