INTERNET-DRAFT
draft-ietf-cat-iakerb-03.txt
Updates: RFC 1510
expires April 30, 2000

Mike Swift
Microsoft
Jonathan Trostle
Cisco Systems

Initial Authentication and Pass Through Authentication Using Kerberos V5 and the GSS-API (IAKERB)

0. Status Of This Memo

1. Abstract

This document defines an extension to the Kerberos protocol specification (RFC 1510 [1]) and GSSAPI Kerberos mechanism (RFC 1964 [2]) that enables a client to obtain Kerberos tickets for services where:

(1) The client knows its principal name and password, but not its realm name (applicable in the situation where a user is already on the network but needs to authenticate to an ISP, and the user does not know his ISP realm name).
(2) The client is able to obtain the IP address of the service in a realm which it wants to send a request to, but is otherwise unable to locate or communicate with a KDC in the service realm or one of the intermediate realms. (One example would be a dial up user who does not have direct IP connectivity).
(3) The client does not know the realm name of the service.

2. Motivation

When authenticating using Kerberos V5, clients obtain tickets from

a KDC and present them to services. This method of operation works well in many situations, but is not always applicable since it requires the client to know its own realm, the realm of the target

service, the names of the KDC's, and to be able to connect to the KDC's.

This document defines an extension to the Kerberos protocol specification (RFC 1510) [1] that enables a client to obtain Kerberos tickets for services where:

(1) The client knows its principal name and password, but not its realm name (applicable in the situation where a user is already on the network but needs to authenticate to an ISP, and the user does not know his ISP realm name).
(2) The client is able to obtain the IP address of the service in a realm which it wants to send a request to, but is otherwise unable to locate or communicate with a KDC in the service realm or one of the intermediate realms. (One example would be a dial up user who does not have direct IP connectivity).
(3) The client does not know the realm name of the service.

In this proposal, the client sends KDC request messages directly to application servers if one of the above failure cases develops. The application server acts as a proxy, forwarding messages back and forth between the client and various KDC's (see Figure 1).

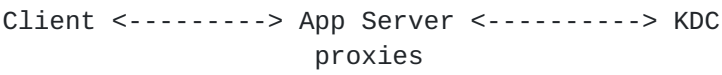


Figure 1: IAKERB proxying

In the case where the client has sent a TGS_REQ message to the application server without a realm name in the request, the application server will forward an error message to the client with its realm name in the e-data field of the error message. The client will attempt to proceed using conventional Kerberos.

3. When Clients Should Use IAKERB

We list several, but possibly not all, cases where the client should use IAKERB. In general, the existing Kerberos paradigm where clients contact the KDC to obtain service tickets should be preserved where possible.

(a) AS_REQ cases:

(i) The client is unable to locate the user's KDC or the KDC's in the user's realm are not responding, or
(ii) The user has not entered a name which can be converted into a realm name (and the realm name cannot be derived from a certificate).

(b) TGS_REQ cases:

(i) the client determines that the KDC(s) in either an intermediate realm or the service realm are not responding or the client is unable to locate a KDC,

(ii) the client is not able to generate the application server realm name.

4. GSSAPI Encapsulation

The mechanism ID for IAKERB 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 AS request, AS reply, TGS request, and TGS reply messages are all encapsulated using the format defined by RFC1964 [2]. This consists of the GSS-API token framing defined in appendix B of RFC1508 [3]:

```
InitialContextToken ::=
[APPLICATION 0] IMPLICIT SEQUENCE {
        thisMech        MechType
                -- MechType is OBJECT IDENTIFIER
                -- representing "Kerberos V5"
        innerContextToken ANY DEFINED BY thisMech
                -- contents mechanism-specific;
                -- ASN.1 usage within innerContextToken
                -- is not required
        }
```

The innerContextToken consists of a 2-byte TOK_ID field (defined below), followed by the Kerberos V5 KRB-AS-REQ, KRB-AS-REP, KRB-TGS-REQ, or KRB-TGS-REP messages, as appropriate. The TOK_ID field shall be one of the following values, to denote that the message is either a request to the KDC or a response from the KDC.

Message	TOK_ID
KRB-KDC-REQ	00 03
KRB-KDC-REP	01 03

5. The Protocol

a. The user supplies a password (AS_REQ): Here the Kerberos client will send an AS_REQ message to the application server if it cannot locate a KDC for the user's realm, or such KDC's do not respond, or the user does not enter a name from which the client can derive the user's realm name. The client sets the realm field of the request equal to its own realm if the realm name is known, otherwise the realm length is set to 0. Upon receipt of the AS_REQ

message, the application server checks if the client has included a realm.

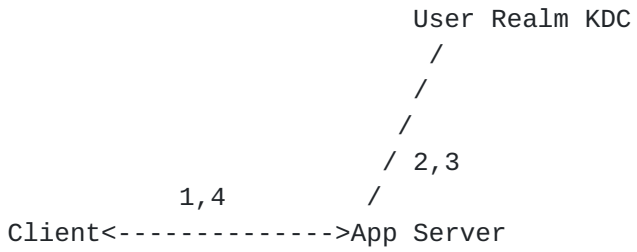
If the realm was not included in the original request, the application server must determine the realm and add it to the AS_REQ message before forwarding it. If the application server cannot determine the client realm, it returns the KRB_AP_ERR_REALM_REQUIRED error-code in an error message to the client):

```
      KRB_AP_ERR_REALM_REQUIRED         77
```

The error message can be sent in response to either an AS_REQ message, in which case the e-data is empty; or in response to a TGS_REQ message, in which case the e-data will contain the realm of the application server encoded as an OCTET STRING. Once the realm is filled in, the application server forwards the request to a KDC in the user's realm. It will retry the request if necessary, and forward the KDC response back to the client.

At the time the user enters a username and password, the client should create a new credential with an INTERNAL NAME [3] that can be used as an input into the GSS_Acquire_cred function call.

This functionality is useful when there is no trust relationship between the user's logon realm and the target realm (Figure 2).



1 Client sends AS_REQ to App Server
2 App server forwards AS_REQ to User Realm KDC
3 App server receives AS_REP from User Realm KDC
4 App server sends AS_REP back to Client

Figure 2: IAKERB AS_REQ

b. The user does not supply a password (TGS_REQ): The user includes a TGT targetted at the user's realm, or an intermediate realm, in a TGS_REQ message. The TGS_REQ message is sent to the application server.

   If the client has included the realm name in the TGS request, then the application server will forward the request to a KDC in the request TGT srealm. It will forward the response back to the client.

If the client has not included the realm name in the TGS request, then the application server will return its realm name to the client using the KRB_AP_ERR_REALM_REQUIRED error described above. The error message e-data field contains the application server realm name. Note that another principal with the same principal name and a different realm than the intended application server can replace the realm name with its own, thus setting the stage for an impersonation attack. Therefore, sending a TGS_REQ message to the application server without a realm name in the request is a last resort (see security considerations below).

The client can now proceed using conventional Kerberos with the realm name from the error message.

6. 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 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.

7. 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 [4]). In such 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.

8. Security Considerations

A principal is identified by its principal name and realm. A client that sends a TGS request to an application server without the request realm name will only be able to mutually authenticate the server up to its principal name; another server with the same principal name and a different realm name, that has a trust relationship with the client, will be able to impersonate the intended application server. Thus, such requests should only be used as a last resort.

9. Bibliography

[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] J. Linn. Generic Security Service Application Program Interface. Request for Comments 1508

[4] B. Tung, C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, J. Wray, J. Trostle, Public Key Cryptography for Initial Authentication in Kerberos, http://www.ietf.org/internet-drafts/draft-ietf-cat-kerberos-pkinit-10.txt.

10. Expires April 30, 2000.

11. Authors' Addresses

Michael Swift
Microsoft
One Microsoft Way
Redmond, Washington, 98052, U.S.A.
Email: mikesw@microsoft.com

Jonathan Trostle
170 W. Tasman Dr.
San Jose, CA 95134, U.S.A.
Email: jtrostle@cisco.com
Phone: (408) 527-6201