Anonymous Credentials in Kerberos

0. 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).

The distribution of this memo is unlimited. It is filed as draft-ietf-cat-kerberos-anoncred-01.txt, and expires December 31, 1997. Please send comments to the authors.

1. Abstract

This document defines the concept of anonymous Kerberos credentials, and describes how such credentials can be securely obtained from a Kerberos KDC via the PKINIT extension. This draft defines no new mechanisms or protocols; instead, it defines the concepts and proposes usage and naming conventions.

2. Introduction

In the Kerberos system[1], the establishment of a secure client-server channel requires both parties to be registered with an authentication service (e.g., a Kerberos realm or a certification service ala PKINIT[2]). In an environment where users are transitory (e.g. a public terminal room or an anonymous ftp site), advance user registration may not be a viable option, yet protection against passive and active attacks is still needed.

Similarly to SSH and SSL, Kerberos should facilitate a way to establish an encrypted channel between a server and an anonymous user; this can be accomplished using anonymous credentials, as described in Section 3. Additionally, the approach presented in this draft enables users who are registered in a Kerberos realm to establish secure, anonymous sessions (e.g., for anonymous e-payment transactions[3]).
3. Framework for Anonymous Credentials

An anonymous ticket is identical to a regular Kerberos ticket defined in RFC 1510. The only difference is that the client principal name, specified in the ticket, is not assigned to any user in a Kerberos realm, nor is there an entry for that name in the Kerberos database. The particular anonymous name will be configurable on a per-realm basis (e.g., anonymous@ISI.EDU or nobody@ISI.EDU for the ISI.EDU realm).

An anonymous ticket can be a ticket granting ticket (TGT) or an end service ticket. A user, in possession of an anonymous ticket and the corresponding session key, can establish an encrypted channel (resilient to passive and active attacks) with the server specified in the ticket.

We propose two methods for obtaining an anonymous ticket from the KDC:

1) In the first method, the user does not share a secret with a KDC or posses a public key certificate. The challenge, is to securely deliver to the client the session key associated with the ticket. Without any modifications, we can utilize the PKINIT extension to Kerberos to achieve this goal. PKINIT employs public key cryptography to obtain a standard ticket granting ticket. In PKINIT the AS-REQ and AS-REP remain the same (per RFC 1510); all PK operations take place in the pre-authentication structure (see [2] for details). PKINIT section 3.2 employs Diffie-Hellman to establish a shared secret which is then used to protect the session key returned in the AS-REQ. In this option, both the client and the KDC authenticate their respective DH public values by signing with a private key.

To obtain anonymous credentials, we propose that the user performs a NULL signature over the DH public value. This is basically a no-op operation which is legal according to the PKINIT specification. The client also sets a new flag, ANONYMOUS_REQUEST in the kdc-options field of KRB_KDC_REQ (see 5.4.1 of [4]).

Upon receiving the request, the KDC creates an anonymous ticket (for the TGT service or for an end service, depending on server principal name requested) and returns in the AS-REP with the corresponding preauth data type (PA-PK-AS-REP). If it was an anonymous TGT, then a client may use it to obtain an anonymous end service ticket using a standard TGT-REQ.

2) The second method enables users already registered in a Kerberos realm to obtain anonymous credentials. A client simply makes a standard AS-REQ or TGS-REQ with the ANONYMOUS_REQUEST flag set. The KDC returns an anonymous ticket. Thus, users that wish to remain anonymous to an application service can setup a secure channel without incurring the cost of PK operations (see case 1). It is important to note that, in the second method, the Kerberos server is trusted to not record and later reveal the principal name of the client that obtained the anonymous ticket.
4. Discussion

We solicit discussion on the implications of the following aspects of the proposal:

ANONYMOUS_REQUEST flag: The use of the ANONYMOUS_REQUEST flag raises issues regarding its propagation. The flag is set by the client in the AS-REQ or a TGS-REQ. Should it be propagated to tickets, and if so, under what conditions?

Services generally should not need to know that clients are anonymous. Authentication and authorization are completely distinct operations. The fact that a user is anonymous rather than well-known should not be important to services, since they should not be basing authorization decisions merely on the fact that the client has a service ticket; any server that does so is arguably broken. Thus propagation of the ANONYMOUS_TICKET flag into service tickets need not be mandatory, and servers may be unaware of a user’s anonymity.

However, there are cases where a service may need to know whether a client principal is anonymous or well-known. For example, consider a service that allows all users access but wishes to maintain an audit log of all actions performed and on whose behalf they are performed. Such a service might want to deny service to anonymous users, not because they are not authorized, but because they are not auditable. The need for such detection is an argument for mandatory propagation of the flag into all service tickets.

We solicit discussion on whether ANONYMOUS_TICKET-flag propagation behavior should be mandated, or whether it should be configurable on a per-realm basis.

5. Bibliography


6. Acknowledgements

Some of the ideas contained in this proposal are based on suggestions by Clifford Neuman, offered in the course of developing the ideas for NetCash[3].

7. Authors
Ari Medvinsky
Matt Hur
Jon Cargille
CyberSafe Corporation
1605 NW Sammamish Road Suite 310
Issaquah WA 98027-5378
Phone: +1 425 391 6000
E-mail: (jonathan.cargille, matt.hur, ari.medvinsky)@cybersafe.com