Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2014

N. Williams
Cryptonector

July 11, 2013

Public Key-Based Kerberos Cross Realm Path Traversal Protocol
draft-williams-kitten-krb5-pkcross-01

Abstract

This document specifies a protocol for obtaining cross-realm Kerberos tickets using existing, related protocols. The resulting protocol has a number of desirable security properties, including privacy protection for the user relative to their home realm's infrastructure, as well a support for leap-of-faith trust establishment, and automated cross-realm keying. This protocol allows Kerberos to scale to large numbers of realms.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 12, 2014.

Table of Contents

1. Introduction

Kerberos [RFC4120] supports meshes of many realms. The individual relationships between realms must be manually keyed, usually with keys derived from passwords. These keys are very difficult to rollover, and when they are changed the result is often outages -- controlled outages where foreseen, but outages nonetheless. This method of cross-realm keying does not scale, and has very poor security properties. We seek to remediate this.

Many years ago there was a proposal for exchanging cross-realm keys using a public key infrastructure (PKI) [RFC5280]; that proposal went by the name "PKCROSS". We appropriate that long-dead proposal's name, but the protocol specified here is very different from the original proposal.

1.1. 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 [RFC2119].

2. The Protocol

A Kerberos client in with a ticket-granting ticket (TGT) for any one source realm (usually but not necessarily the client's own realm) wishing to acquire a TGT for a destination realm may use this protocol instead of the traditional cross-realm ticket-granting service (TGS) exchanges as follows:

1. Generate private key to a public key cryptosystem;

2. Generate a certificate signing request (CSR) [RFC2986], such that the resulting certificate has an id-pkinit-san subject alternative name (SAN) corresponding to the client's principal name and realm;

3. Request a certificate from the kx509 [I-D.hotz-kx509] service run by the source realm;

4. Request a TGT from the destination realm using PKINIT [RFC4556].

If the destination realm issues the requested Ticket then it SHOULD include the client's certificate in an AD-CLIENT-CERTIFICATE authorization-data element, and it MUST do so if it does not validate the client's certificate to an acceptable trust anchor.

The destination realm MUST NOT set the TRANSIT-POLICY-CHECKED flag on the tickets they issue to clients whose foreign realm certificates are not validated by the KDC. Destination realm administrators may configure their realms to know specific foreign realm clients' certificates.

The destination MUST include the trust path of the client's certificate, if validated, in the 'transited' field of the issued Ticket, using a mapping of the issuer names to the X.500 realm naming style [XXX must specify this mapping; hopefully it can be the identity function or close enough].

2.1. Exchange of Long-Term Cross-Realm Symmetric Keys

When the client principal is a TGS principal and its PKINIT AS-REQ protocol data unit (PDU) has the USE-SESSION-KEY-AS-REALM-KEY KDCOptions flag set then the client is requesting that the session key of the ticket issued by the destination realm become the long-term key for the corresponding krbtgt/DESTINATION@SOURCE principal. The destination realm MUST validate the client principal's certificate, building a trust path if need be, and validating it to a trust anchor. The source and destination realm MAY have previously exchange fingerprints of their respective key distribution service

(KDC) public keys and/or certificates and/or the source realm's kx509 root or intermediate certification authority (CA), and such previously exchanged material, if any, MUST be used for certificate trust validation.

Realm administrators should use the procedure to setup symmetric cross-realm keys as necessary to save clients from having to frequently use kx509 and PKINIT as described in the preceding section.

Where public key infrastructure (PKI) exists allowing this to happen automatically, realms' KDCs MAY be configured to automatically key cross-realm principals for any realms that their source realms' clients request cross-realm TGTs for, but note that this presents a denial of service (DoS) opportunity to the source realm's clients. Source realm KDCs SHOULD only do this when a) they are configured to do so, b) the requesting client principal is in the same realm, c) the KDC has not spent too much effort recently providing this service (i.e., KDCs should throttle attempts to establish symmetric cross-realm keys in this manner), and d) up to some maximum number of cross-realm principals.

3. Privacy Protection

This protocol protects the privacy of client principals vis-a-vis their home realms: client principals' home realms need not know what destination realms the clients are speaking to because client principals need not ask their home realms.

4. Leap-of-Faith / TOFU

Clients need not validate the certificate trust path of destination realms. When they do not, the services used through those destination realms are as good as anonymous authentication. If the client saves the root or intermediate or end entity certificates of the destination realms that it cannot or does not validate, then the client can check that on future occasions the destination realm's certificate has not changed, and it may warn the user if it has. This quite similar to how clients using the secure shell (SSH) protocol [RFC4251] handle server authentication, and is commonly known as "leap-of-faith" (LoF) or trust-on-first-use (TOFU). The result is as good as pseudonymous authentication.

Destination services too may apply apply LoF/TOFU: by not validating the transit path of the client (e.g., if it's not in a white-list of realms whose clients must have valid transit paths) and accepting tickets without the TRANSITED-POLICY-CHECKED ticket flag set. The destination service can save the client's certificate, if found in an AD-CLIENT-CERTIFICATE authorization-data element in the client's Ticket, and may use it later to ensure that it is talking to the same client.

5. Using DNSSEC for Realm Certificate Validation

[Specify how to use DNS-Based Authentication of Named Entities (DANE) [RFC6698] to authenticate the KDC certificates of realms with domain-style names. Roughly: format the realm's name as a domainname, then format the DANE TLSA resource record set's (RRset) domainname per-DANE, using the KDC's port number. Note that the KDCs will usually not speak TLS, though there is an extension for using TLS in the KDC over TCP protocol. For example, the TLSA RRset for any KDC for the DESTINATION.EXAMPLE realm might be named _88._tcp.destination.example.]

6. Security Considerations

[[anchor1: ...]]

7. IANA Considerations

[[anchor2: Allocate the new KDCOptions flag (USE-SESSION-KEY-AS-REALM-KEY) and authorization-data element (AD-CLIENT-CERTIFICATE).]]

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, November 2000.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, August 2012.

[I-D.hotz-kx509] Hotz, H. and R. Allbery, "KX509 Kerberized Certificate Issuance Protocol in Use in 2012", draft-hotz-kx509-06 (work in progress), July 2012.

8.2. Informative References

[RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", RFC 4251, January 2006.

Author's Address

Nicolas Williams
Cryptonector, LLC

Email: nico@cryptonector.com