Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 14, 2014

N. Williams
Cryptonector

August 13, 2013

Public Key-Based Kerberos Cross Realm Path Traversal Protocol Using Kerberized Certification Authorities (kx509) and PKINIT
draft-williams-kitten-krb5-pkcross-02

Abstract

This document specifies a protocol for obtaining cross-realm Kerberos tickets using existing, related protocols: kerberized certification authorities (kx509) and public key cryptography initial authentication in Kerberos (PKINIT). 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 February 14, 2014.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Conventions used in this document 3
2. The Protocol 4
2.1. Exchange of Long-Term Cross-Realm Symmetric Keys 4
3. Security Properties 6
3.1. Automated Cross-Realm Keying 6
3.2. Privacy Protection relative to home realm 6
3.3. Leap-of-Faith (LoF) / Trust-On-First-Use (TOFU) 6
3.3.1. Requirements and Recommendations for LoF/TOFU Authentication 6
4. Using DANE (DNSSEC) for Realm Certificate Validation . . . 8
5. Application Programming Interface Considerations 9
5.1. API Considerations for LoF/TOFU Authentication 9
5.2. GSS-API Naming Considerations 9
6. Security Considerations 10
6.1. Loss of Cross-Realm Principal Trust Establishment Information 10
6.2. Security Considerations for LoF/TOFU 10
6.3. On the Need for a Common Transit Path Policy Language . . 11
7. IANA Considerations 12
8. References 13
8.1. Normative References 13
8.2. Informative References 13
Author's Address 14

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 [RFC6717] 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. Security Properties

The proposed PKCROSS protocol has several useful properties described below.

3.1. Automated Cross-Realm Keying

No more manual keying of cross-realm principals via exchanging passwords on a telephone call (or similar).

3.2. Privacy Protection relative to home realm

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.

This feature is generally and naturally available in PKI, and as this protocol is based on a kerberized certification authority, this protocol inherits this privacy feature from PKI.

3.3. Leap-of-Faith (LoF) / Trust-On-First-Use (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 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.

3.3.1. Requirements and Recommendations for LoF/TOFU Authentication

- Implementations MUST NOT use LoF/TOFU to authenticate a target service's realm without the approval of the user or without making it clear that the realm is not fully authenticated (perhaps by replacing the realm's name with a fingerprint of its public key / certificate).

- Implementations MAY allow service administrators to establish user-friendly aliases for client principal names that include public key fingerprint material.

- Implementations MAY provide a way to automatically learn realm name <-> public key / certificate bindings. Pinning [add reference to HSTS] SHOULD be supported in that case. The user MUST approve of each such mapping.

4. Using DANE (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.]

5. Application Programming Interface Considerations

For non-LoF/TOFU uses the main security consideration for applications is that improved scalability for Kerberos realm traversal implies larger Kerberos universes, and the larger a universe of trust the more important it is to have useful and expressive local policy for evaluating the trustworthiness of any given transit path. Because in most applications local policy should be a component external to the application, there is little impact on APIs here. However, an implementation may wish to provide applications with interfaces for specifying policies, either named or by value.

5.1. API Considerations for LoF/TOFU Authentication

For LoF/TOFU uses there is a critical requirement that APIs not permit accidental aliasing of principal names as a result of LoF/TOFU being used. The simplest way to do this is to use a fingerprint of the peer principal's public key as their principal, and/or a fingerprint of the peer principal's realm's public key as their realm.

[[anchor1: For interoperability and compatibility we ought to specify what fingerprint algorithm to use, perhaps one of the SSHv2 fingerprint algorithms, such as in RFC4255, but those use weaker hashes...]]

5.2. GSS-API Naming Considerations

There are no GSS-API-specific considerations. The naming considerations described in Section 5.1 and the naming attributes defined in [I-D.williams-kitten-generic-naming-attributes] are sufficient. Note however that information about how PKCROSS was used to establish symmetrically-keyed cross-realm principals is lost and will not appear in the transit path in tickets issued by KDCs reached via such cross-realm principals.

[[anchor2: Actually, we may need to specify some interfaces by which to indicate that the user wishes to alias a pseudonymous name. Perhaps we can do so by applying GSS_Set_name_attribute() to a peer MN obtained from GSS_Inquire_context()?]]

6. Security Considerations

[[anchor3: All the security considerations of Kerberos and PKI apply. Security considerations are discussed throughout this document.]]

Scaling up the universe of realms reachable via any trust path necessarily dilutes trust overall, but not for specific paths. On the other hand, by shortening transit path lengths trust can be improved, though some short transit paths will have been symmetrically keyed using this PKCROSS protocol and therefore will be longer than they appear to be. These are subjective notions of trust, of course.

6.1. Loss of Cross-Realm Principal Trust Establishment Information

Note that once a cross-realm principal is symmetrically keyed no information about how that keying operation took place will appear in tickets issued by that TGS principal.

Note also that the Kebreros transit path encodes only realm names (including X.500-style names, thus PKIX certificate subject and issuer names), and lacks any public key information that might be useful for pinning. However, the certificate validation path for each realm in a transit path SHOULD be included in the transit path.

6.2. Security Considerations for LoF/TOFU

LoF/TOFU has additional security considerations. To start there is the obvious susceptibility to peer impersonation / man-in-the-middle (MITM) attacks on initial contact, which is mitigated by the attacker's need to always remain in the middle in order to avoid detection.

LoF/TOFU require the ability to remember peers' pseudonymous identities -- their public keys (or certificates), otherwise one remains vulnerable to peer impersonation / MITM attacks at all times. This requires synchronization of peer pseudonym databases across multiple devices (where users have multiple devices), which may not always be possible or performed.

It is critical that existing applications not be broken by the ability to use LoF/TOFU in new Kerberos implementations when those applications are re-linked with newer Kerberos implementations. To ensure this we require the use of public key fingerprints as principal and/or realm names; local mappings of learned pseudonym mappings onto semantically meaningful names are permitted where the user can validate the mapping. But keep in mind that most users never actually do much to verify peers' public keys in any

application/protocol that provides LoF/TOFU [references for this would be nice -Nico].

See Section 3.3.1 for additional requirements for LoF/TOFU authentication.

6.3. On the Need for a Common Transit Path Policy Language

There are no standard ways to express authorization policies for trust transit paths for either Kerberos nor PKI. A standard language for this would be extremely useful. Such a language should allow for the expression of policies for both, clients and services. Such a language should allow for the expression of complex realm/domain/other naming, and should allow for HSTS-style pinning [add references -Nico]. Such a language should allow for multiple paths where desired, and should allow for more than path rejection: it should also allow for reducing the entitlements assigned to a peer/realm for authorization purposes.

The need for a standard transit path policy expression language is not new, and such a language is broadly and generally needed. Therefore such a language is outside this document's scope.

7. IANA Considerations

[[anchor4: 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.

[RFC6717] Hotz, H. and R. Allbery, "kx509 Kerberized Certificate Issuance Protocol in Use in 2012", RFC 6717, August 2012.

[I-D.williams-kitten-generic-naming-attributes] Williams, N., "Generic Naming Attributes for the Generic Security Services Application Programming Interface (GSS-API)", draft-williams-kitten-generic-naming-attributes-00 (work in progress), July 2013.

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