Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2015

N. Williams
Cryptonector

October 27, 2014

Public Key-Based Kerberos Cross Realm Path Traversal Protocol Using Kerberized Certification Authorities (kx509) and PKINIT
draft-williams-kitten-krb5-pkcross-05

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 properties, primarily that it 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 April 30, 2015.

Copyright Notice

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. A full mesh wouldn't scale, therefore the protocol calls for hierarchical trust universes. In practice non-hierarchical but also non-fully-meshed relationships are used, and these generally require distribution of trust routing information to clients, services, and KDCs. With referrals it is possible to reduce the need for client-side trust routing information, but KDCs still need it, as do services (unless they accept KDC trust path policy and the KDC applies it via the TRANSITED-POLICY-CHECKED ticket flag).

These manually-exchanged keys are very difficult to rollover safely, and when they are changed the result is often outages -- controlled outages where foreseen, but outages nonetheless.

Manual cross-realm keying does not scale, and has very poor security properties. We seek to remediate this using public key cryptography, building on existing Kerberos specifications.

Distribution of trust routing (traditionally known as "capaths") and trust path validation (also "capaths") information is difficult; there is no standard protocol for it. Maintenance of it is a thoroughly manual process.

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.

PKCROSS can make Kerberos scale to large numbers of realms, will remove the need for manual keying of cross-realm TGS principals, will further reduce the need for maintenance and distribution of trust

routing information, and will tend to reduce the complexity of trust path validation.

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 PKCROSS Protocol

We provide two variants of the PKCROSS protocol: one that is client-driven, and another that is driven by a Ticket Granting Service (TGS) on behalf of its clients. The latter is based on the former, with the TGS acting as a client. We begin with the client-driven case. DNS-Based Authentication of Named Entities (DANE) [RFC6698] can and should be used for realm CA certificate validation.

2.1. Client-Driven PKCROSS

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. Request a certificate from the kx509 [RFC6717] service run by the source realm;

3. Request a TGT from the destination realm using PKINIT [RFC4556] and the client certificate obtained in step #2.

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 AD-

CLIENT-CERTIFICATE authorization-data MUST be in a KDC-signed authorization-data container [XXX add reference to CAMMAC].

[[anchor1: QUESTION: Should the PKINIT request in step #3 be a TGS-REQ with PKINIT pre-auth data?]]

[[anchor2: QUESTION: Should the PKINIT request in step #3 be required to be used within a FAST tunnel?]]

2.2. TGS-Driven PKCROSS

A TGS can bootstrap ephemeral cross-realm trust principals on behalf of its clients. This allows the cost of PKCROSS to be amortized over many clients, and it allows participation by clients that do not support client-driven PKCROSS (or whose PKCROSS requests are rejected by the target).

In this mode the TGS uses the client-driven PKCROSS protocol, modified as follows:

- the TGS's client certificate MUST have an id-pkinit-san Subject Alternative Name (SAN) identifying the source TGS as krbtgt/SOURCE@SOURCE
- the TGS's client certificate MUST have an Extended Key Usage (EKU) of id-pkcross-issuer (TBD)

The resulting TGT -which we shall term an "issuer TGT" (ITGT)- and its session key can then be used by the source TGS to create cross-realm TGTs for the source-to-target trust principal ("krbtgt/TARGET@SOURCE").

This ITGT will be used to mint tickets as described below.

2.2.1. Issuing cross-realm TGTs issued for PKCROSS-keyed cross-realm TGS principals

Cross-realm TGTs issued by a source TGS using an ITGT will not be quite like normal Kerberos Tickets: their encrypted part contains an AP-REQ using the ITGT acquired by the source TGS, and this AP-REQ is "encrypted" with the null enctype, The AP-REQ's Authenitcator MUST contain an authorization-data element that carries a) the name of the

client principal, b) the session key that the client should be using with the cross-realm TGTs issued.

```
 AD-PKCROSS-TGT-INFO ::= SEQUENCE {
     cname [0] Principal,    -- the client's realm is the
                             -- crealm from the ITGT's EncTicketPart
     key   [1] EncryptionKey
 }
```

Figure 1: AD-PKCROSS-TGT-INFO

2.2.2. Handling impatient clients

Because the process of acquiring an ITGT might be slow, a TGS doing so on behalf of a client could use a mechanism for instructing the client to be patient. Existing clients would not handler a new error code by waiting, therefore there is not much that can be done to keep an impatient client from retrying at another KDC.

The existing KDC_ERR_SVC_UNAVAILABLE error code cannot be used as often this causes the client to immediately retry the request at another KDC. A new error code for indicating estimated time to completion of request would be handy, but out of scope for this document.

Note that there is a denial of service (DoS) attack by clients on

willing source KDCs: the clients can ask the KDCs to acquire cross-realm ITGTs for many target realms. Ideally the quality of service for the Kerberos authentication service (AS) with PKINIT (and/or other slow pre-authentication mechanisms) should be separate from that of the Kerberos TGS co-located with it, and the PKCROSS-capable TGS as well, so as to be able to throttle low-priority requests when under load.

2.3. Stapled DANE

[[anchor3: TBD. We should use Google's serialization of DNS RRsets needed for DANE validation. We will need a label for the TLSA RRs for kx509 issuers.]]

2.4. Validation

KDCs processing PKINIT requests crossing realms MUST apply either or both of:

- PKIX certificate validation
- DANE certificate validation

KDCs MUST reject PKINIT requests from clients of foreign realms whose certificates cannot be validated, unless the client request the anonymous principal name in the target's realm.

2.5. Transit Path

The combined Kerberos/PKIX/DNSSEC transit path MUST be represented in any tickets issued using PKCROSS (see below). As usual, each realm's KDCs in the mix can set the transit policy checked flag if a client's transit path is acceptable per the realm's KDCs' local policy.

Two validation mechanisms are available: all PKIX [RFC5280] validation methods, and DANE [RFC6698]. DANE validation records SHOULD be stapled onto the client certificates by the issuing kx509 CA; alternatively, clients can staple <http://src.chromium.org/viewvc/chrome/trunk/src/net/base/dnssec_chain_verifier.cc?pathrev=167227> onto their PKINIT requests using an authorization-data element, AD-PKINIT-CLIENT-DANE.

Additionally, when PKIX certificate validation is used, the trust path should be encoded in an AD-INITIAL-VERIFIED-CAS authorization data element, per-PKINIT.

2.5.1. Transit path representation

The notional transit path for a ticket issued by a target realm's KDCs includes:

- the source realm (never expressed in the 'transited' field of Kerberos Tickets)

- all realms in the ITGT's transited field (in the TGS-driven PKCROSS case)
- all issuers in the validation path for the kx509-issued certificate, which are
  - all issuers in the certificate's PKIX validation path when PKIX validation is used
  - all DNS zone domainnames transited from the source realm's domainname to the root zone
- the target realm (also never expressed in the 'transited' field)

When using DANE for validation of the issuer's certificate the target SHOULD represent the transit path as hierarchical from the source realm's domain to the root domain, then direct from there to the target's realm.

The notional transit path for a given client principal MUST be encoded as usual, using the Kerberos X.500 and domain-style representations of PKIX issuer names and DNS domainnames as faithfully to the original as possible.

[[anchor4: QUESTION: Do we need a 100% faithful representation of the transit path?]]

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

A KDC can acquire a TGT using PKCROSS whose session key then becomes the long-lived, persistent symmetric key for a cross-realm principal from the source realm to the target realm ("krbtgt/TARGET@SOURCE").

To do this the KDC MUST set the USE-SESSION-KEY-AS-REALM-KEY KDCOptions flag (TBD) in its request for an ITGT from the target realm. As usual, the target realm's KDC MUST validate the client principal's certificate. The target realm's KDC MUST NOT return a TGS-REP until the new principal is committed to its principal database, and MUST set the endtime of the ITGT to the time at which the source realm may begin using the new symmetrically-keyed

principal.

The source realm's KDC MUST commit the new principal to its principal database and MUST NOT begin using the new principal's long-term keys until the new principal is available to all KDCs for the source realm and the endtime of the ITGT passes.

Target KDCs SHOULD require manual pre-approval of such new cross-realm principals. In small, isolated environments a KDC MAY be configured to pre-approve all such new principals.

By default, source KDCs SHOULD NOT automatically request long-term keying of cross-realm principals.

3. Security Properties

The proposed PKCROSS protocol has several useful properties described below.

3.1. Automatic Cross-Realm Keying

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

3.2. Scalability

Kerberos with commonplace symmetrically-keyed hierarchical cross-real trusts can scale to a large universe of realms, but only if there are top-level realms that are willing to pair-wise trust and "child" realms. Such top-level realms do not exist in practice, leading to an O(N^2) scaling problem for most two-label realms.

Leveraging a PKI, such as a PKIX PKI [RFC5280] or a DNSSEC PKI [RFC4033] removes the need for either top-level realms (which are not likely to ever be operated as commercial or even non-profit entities) or O(N^2) pair-wise cross-realm symmetric keying.

The cost of this is having to add PKI trust paths to Kerberos trust paths (though the resulting trust path length need not be much different than before).

3.2.1. Simplified trust routing

For clients, relying on referrals (and TGS-driven PKCROSS) and/or client-driven PKCROSS will greatly reduce the need for client-side trust routing information.

Even KDCs won't need trust routing information.

3.2.2. Simplified trust path validation

For services that accept hierarchical trust paths, PKCROSS will greatly reduce the complexity of trust path / transit validation. Such services that also trust DANE/DNSSEC will need no trust path validation information for any clients using PKCROSS to reach the service. In many cases a very simple policy expressed in terms of whitelists of top-level domains (TLDs) or near top-level domains traversed, trust anchor sets, and trust of all zero-length transit paths, will suffice.

3.3. Privacy Protection relative to home realm

This protocol protects the privacy of client principals vis-a-vis their home realms, when the clients use the client-driven PKCROSS protocol.

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.

The realms visited by the client may, of course, inform the client's home realm, but in the event that they don't, the client does gain this small measure of privacy. Of course, the privacy-conscious client SHOULD attach an OCSP Response [RFC6960] to its PKINIT request, per [RFC4557].

4. Application Programming Interface Considerations

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 mostly no impact on APIs here. However, an implementation may wish to provide applications with interfaces for specifying policies, either named or by value.

4.1. GSS-API Considerations

The naming attributes [RFC6680] defined in [I-D.williams-kitten-generic-naming-attributes] provide access to information about transit paths.

Note 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.

5. Security Considerations

[[anchor5: 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.

5.1. Loss of Cross-Realm Principal Trust Establishment Information

Once a cross-realm principal is symmetrically keyed the transit path used to automatically key that principal will no longer appear in subsequent cross-realm tickets issued by the target.

The Kerberos 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.

5.2. 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.

PKCROSS can greatly simplify the process of validating a ticket's trust path, first by shortening the number of realms involved to two (in the typical case), maybe three, second by making the actual trust path (PKI or DANE) hierarchical, thus hopefully leaving much less policy to express in a transit path policy language: whitelists of domains and sub-domains, perhaps. But a common language would still

be desirable.

5.3. On the Need for Trust Routing

A common language for trust routing is not necessary in a purely hierarchical world, as in DANE. But since it's likely that there will be some non-hierarchical, non-zero-length transit paths in many deployments for a long time to come, a common language for trust routing would be desirable as well. Routing protocols normally used for network addresses could be used for discovery and distribution of trust routing information as well. But note that there are subtle differences between trust routing and trust path validation, even though in traditional Kerberos deployments the same information is used for both, with the trust validation policy effectively being that the client must have taken the shortest, highest-priority path specified in "capaths" configutation.

6. IANA Considerations

[[anchor6: Allocate the new KDCOptions flag (USE-SESSION-KEY-AS-REALM-KEY) and authorization-data element (AD-CLIENT-CERTIFICATE), as well as the new EKU id-pkcross-issuer.]]

7. TODO

- Provide a normative reference for DANE stapling.

8. Acknowledgements

Although the author arrived at this "kx509 + PKINIT == PKCROSS" idea independently, it is not an original idea. Henry Hotz and Jeffrey Altman each conceived the same idea years earlier. It is a relatively obvious idea when taking into account efforts to bridge disparate security mechanisms and credentials infrastructures.

9. References

9.1. Normative References

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

[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.

[RFC4557] Zhu, L., Jaganathan, K., and N. Williams, "Online Certificate Status Protocol (OCSP) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4557, 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.

[RFC6680] Williams, N., Johansson, L., Hartman, S., and S. Josefsson, "Generic Security Service Application Programming Interface (GSS-API) Naming Extensions", RFC 6680, August 2012.

[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-01

(work in progress), August 2013.

9.2. Informative References

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, June 2013.

Author's Address

Nicolas Williams
Cryptonector, LLC

Email: nico@cryptonector.com