Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 12, 2011

H. Hotz
Jet Propulsion Laboratory,
California Institute of
Technology
June 10, 2011

KX509 Kerberized Certificate Issuance Protocol
draft-hotz-kx509-03.txt

Abstract

This rfc describes a protocol, called kx509, for using Kerberos tickets to acquire X.509 certificates. These certificates may be used for many of the same purposes as X.509 certificates acquired by other means, but if a Kerberos infrastructure already exists then the overhead of using kx509 may be much less.

While not (previously) standardized, this protocol is already in use at several large organizations, and certificates issued with this protocol are recognized by the International Grid Trust Federation.

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 December 12, 2011.

Copyright Notice

Table of Contents

1. Introduction

The two primary ways of providing cryptographically secure identification on the Internet are Kerberos tickets [RFC4120], and X.509 [RFC5280] and [X.509] certificates.

In practical IT infrastructure where both are in use, it's highly desirable to deploy their support in a way which guarantees they both authoritatively refer to the same entities. There is already a widely-adopted standard for using X.509 certificates to acquire corresponding Kerberos tickets called PKINIT [RFC4556]. This rfc describes the kx509 protocol for supporting the symmetric operation of acquiring X.509 certificates using Kerberos tickets.

The International Grid Trust Federation [IGTF] supports the use of Short Lived Credential Services [SLCS] as a means to authenticate for resource usage based on other, native identity stores which an organization maintains. X.509 certificates issued using the kx509 protocol based on a Kerberos identity is one of the recognized Credential Services. The certificate profile for that use is outside the scope of this RFC, but is described in [GRID-prof].

In normal operation kx509 can be used after a Kerberos ticket-granting-ticket (TGT) is acquired, which is most likely during user login. First, the client generates a RSA public/private key-pair. Next, using the Kerberos ticket-granting-ticket, it acquires a Kerberos service ticket for the KCA (Kerberized Certificate Authority), and uses this to send the public half of its key-pair. The KCA will decrypt the service ticket, verify the integrity of the incoming packet, determine the identity of the user, and use the session key to send back a corresponding X.509 certificate.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Protocol Data

The protocol consists of a single request/reply exchange using UDP.

Both the request and the reply packet begin with four bytes of version ID information, followed by a DER encoded ASN.1 message. The first two bytes of the version ID are reserved. They MUST be set to zero when sent, and SHOULD be ignored when received. The third and fourth bytes are the major and minor version numbers. The version of

the protocol described in this document is designated 2.0, so the first four bytes of the packet are 0, 0, 2, 0.

Incompatible variations of this protocol MUST use a different major version number.

2.1. Request Packet

The request consists of a version ID, followed by a DER encoded ASN.1 message containing a Kerberos AP_REQ, integrity check data on the request, and public key generated by the client. The ASN.1 encoding is:

```
KX509Request ::= SEQUENCE {
        ap-req OCTET STRING,
        pk-hash OCTET STRING,
        pk-key OCTET STRING,
        client-version VisibleString OPTIONAL
}
```

The ap-req is as described in [RFC4120] Section 5.5.1.

The pk-hash is HMAC using SHA-1 as the underlying hash. All 160 bits are sent. The key used is the Kerberos session key. The data is the 4-byte version ID and the octet string for pk-key.

The pk-key contains a public key. This key and its corresponding private key are generated by the client before contacting the server. Implementations of this protocol MUST support RSA keys, in which case the key is a DER encoded RSAPublicKey as defined in [RFC3447], section A.1.1, and then stored in this octet string in the request. Use of other public-key types is not defined.

The optional client-version is described in Appendix B. It does not seem to be used in practice and should be omitted for that reason.

2.2. Reply Packet

The reply consists of a version ID, followed by a DER encoded ASN.1 message containing an error code, and an optional authentication hash, optional certificate, and optional error text. The service SHOULD return replies of the same version as the request where possible.

```
KX509Response ::= SEQUENCE {
        error-code[0] INTEGER DEFAULT 0,
        hash[1] OCTET STRING OPTIONAL,
        certificate[2] OCTET STRING OPTIONAL,
        e-text[3] VisibleString OPTIONAL
}
```

Although the format of the reply contains independently optional objects, the server MUST only generate replies with one of the following allowed combinations.

	hash	certificate	
error-code	hash		e-text
error-code			e-text

The first case is returned when the server successfully generates a certificate for the user. The certificate is a DER encoded Certificate as defined in [RFC5280] Section A, page 116.

The second case is returned when the server successfully authenticates the user and their key, but is unable for some other reason to generate a certificate.

The third case MAY be returned if the server is unable to successfully authenticate the user and intends to return some unauthenticated information to the client.

The hash on a response is computed using SHA-1 HMAC as for the request. The data that is hashed is the concatenation of the 4-byte version ID at the beginning of the packet, the error-code (if present), and all other optional fields which are present except the hash itself. In other words, the hash is computed on the fields which are present exclusive of the overall ASN.1 wrapping. The e-text MAY be translated into other character sets for display purposes, but the hash is computed on the e-text in its VisibleString representation.

If the e-text contains NUL characters, the client MAY ignore any part of the error message after the first NUL character for display purposes.

As implied by the above table, if the reply does not contain a certificate it MUST contain an error message and a non-zero error code. Conversely, if a certificate is returned then the error code MUST be zero. The server SHOULD NOT send a zero error-code. The client MUST treat a missing error-code as if it were zero.

The defined error codes are as follows:

error-code	Condition	Example
1	Permanent problem with client request	Incompatible version
2	Solvable problem with client request	Expired Kerberos credentials
3	Temporary problem with client request	Packet loss
4	Permanent problem with the server	Internal misconfiguration
5	Temporary problem with the server	Server overloaded

If a client error is returned, the client SHOULD NOT retry the request unless some remedial action is first taken. If error-code 3 is returned, the client MAY retry with other servers before giving up.

If a server error is returned, it is RECOMMENDED that the client retry the request with a different server if one is known.

Since all KCAs serving a Kerberos realm are intended to be equivalent, in accordance with [RFC5280] Section 4.1.2.2, the certificates returned from different KCAs serving the same Kerberos realm MUST NOT contain duplicate serial numbers.

The returned certificate MUST identify the Kerberos client principal from the ap-req in the original KX509Request in the subject of the cert, or in a subjectAltName extension. The identification MUST be unique within the organization's deployed infrastructure. It is RECOMMENDED that a subjectAltName extension be included of type id-pkinit-san as described in [RFC4556] Section 3.2.2. Note that the id-pkinit-san is simply a standard representation of a Kerberos principal, and has no other implications with respect to PKINIT.

Other extensions MAY be added according to local policy.

3. Protocol Operation

Absent errors, the protocol consists of a single request, sent via UDP, and a single reply, also sent via UDP.

There is no special provision for requests or replies which exceed

the allowable size of a UDP packet. Also some implementations have imposed hard size limits which are smaller than a typical UDP MTU, and will limit the use of extensions and the supportable key size. Even without hard limits, if the request or reply exceeds the MTU size of a UDP packet for the infrastructure in use, then the reliability of the exchange will decrease significantly.

For "normal" Kerberos ap-req structures, and "normal" X.509 certificates, this is unlikely unless the Kerberos service ticket contains large amounts of authorization data. For this reason, it is RECOMMENDED that service tickets for the KCA be issued without authorization data. If the KCA performs authorization, it should do so by other means.

Before constructing the request, the client must know the canonical name(s) and port(s) of the server(s) to contact. It MAY determine them by looking up the service's SRV record as described in[RFC2782]. The entry to be used is _kca._udp._realm_, where _realm_ is the Kerberos realm, used as part of the DNS name.

The client must then acquire a service ticket in order to construct the ap-req for the service. The Kerberos service principal name to use for this service has a first component of "kca_service". The second component and the realm of the principal follow normal Kerberos conventions.

When the server receives a request, it MUST make sanity checks including at least the following:

- The AP-REQ can be decoded and is not expired.
- If the request uses cross-realm authentication, then it satisfies the requirements of local policy and [RFC4120] Sections 1.2 and 2.7.
- The request's hash is valid.

The server SHOULD make other sanity checks, such as a minimum public key length, to the extent feasible.

The server MAY decline to respond to an erroneous request. If it does not receive a response a client MAY retry its request, but the client SHOULD wait at least one second before doing so.

The client MUST verify any hash in the reply, and MUST NOT use any certificate in a reply whose hash does not verify. The client MAY display the e-text if the hash is absent or does not verify, but SHOULD indicate the message is not authenticated.

4. Acknowledgements

The original version of kx509 was implemented using Kerberos 4 at the University of Michigan, and was nicely documented in [KX509]. Many thanks to them for their original work, as well as the subsequent updates.

While developing this document I received important corrections and comments from Jeffrey Altman, and Love Hornquist Astrand. I also received many helpful comments and corrections from Doug Engert, Jeffrey Hutzelman, Sam Hartman, Timothy J. Miller, and Chaskiel Grundman. Alan Sill provided the references to the International Grid Trust Federation and its acceptable credential services. Example network traffic was provided by Doug Engert, Marcus Watts, Matt Crawford, and Chaskiel Grundman from their deployments, and was extremely useful for verifying the reality of this specification.

5. IANA Considerations

IANA is requested to add "kca_service" as a GSSAPI/Kerberos/SASL service name for a "Kerberized Certificate Authority".

This service is conventionally run on UDP port 9878, but this memo does not request that IANA standardize the port number.

6. Security Considerations

The only encrypted information in the protocol is that used by Kerberos itself. The considerations for any Kerberized service apply here.

The public key in the request is sent in the clear, and without any guarantees that the requestor actually possesses the corresponding private key. Therefore the only appropriate uses of the returned certificate are those where the identity of the requestor is unimportant, or the subsequent use independently guarantees that the user possesses the private key.

Some information, such as the public key and certificate, is transmitted in the clear but (as the name implies) were generally intended to be publicly available. However their visibility could still raise privacy concerns. The hash is used to protect their integrity.

The policies for issuing Kerberos tickets and X.509 certificates are usually expressed very differently. An implementation of this

protocol should not provide a mechanism for bypassing ticket or certificate policies.

In particular, if the issued certificate can be used with PKINIT, this authentication loop SHOULD NOT bypass policy limits for either X.509 certificates or Kerberos tickets.

X.509 certificates are usually issued with considerably longer validity times than Kerberos tickets. Care should be taken that the issued certificate is not valid for longer than the intended policy should allow. Note that[RFC4556] Section 3.2.3.1 REQUIRES that the lifetime of an issued ticket not exceed the lifetime of the predecessor certificate. By analogy it is RECOMMENDED that the lifetime of an issued certificate not exceed the lifetime of the predecessor Kerberos ticket unless the implications with respect to local policy are clearly understood and allow it.

7. References

7.1. Normative References

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

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[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.

7.2. Informative References

[GRID-prof] "GRID Certificate Profile", March 2008, <http://www.ogf.org/documents/GFD.125.pdf>.

[IGTF] "The International Grid Trust Federation", <http://www.igtf.net/>.

[KX509] Doster, W., Watts, M., and D. Hyde, "The KX509 Protocol", September 2001, <http://www.citi.umich.edu/techreports/reports/citi-tr-01-2.pdf>.

[SLCS] "Short Lived Credential Services", February 2009, <http://tagpma.org/authn_profiles/slcs>.

[X.509] International Telecommunications Union, "Recommendation X.509: The Directory: Public-key and attribute certificate framework", November 2008.

Appendix A. Certificate Cacheing and Deployment Considerations

As noted in the Security Considerations section, the functional lifetime of the acquired X.509 certificate should usually match the lifetime of its predecessor Kerberos ticket. Therefore, it is likely that X.509 certificates issued with this protocol should be deleted when the supporting Kerberos tickets are deleted. That makes the Kerberos ticket cache a reasonable location to store the certificate (and its private key).

On the other hand applications, such as web browsers, probably expect certificates in different stores.

A widely used solution to this dichotomy is to implement a PKCS11 library which supports the KX509-acquired credentials. The credentials remain stored in the kerberos credentials cache, but full PKI functionality is still available via a standard interface for PKI credentials.

Appendix B. Historically Used Extensions

A subjectAltName othername extension of type kcaAuthRealm (OID value 1.3.6.1.4.1.250.42.1) is frequently used to include the client's realm as an ASN.1 octet string.

The Microsoft-defined userPrincipalName has frequently been used for

the same purpose as the id-pkinit-san.

As defined in section 2.1, the historic implementations contained code for an additional client-version string in the request. If present, the KCA copied it into the issued certificate as an extension with the oid 1.3.6.1.4.1.250.42.2.

Appendix C. Issues and Changes from the Previous Draft

RFC Editor Note: Delete this appendix before final publication.

Possible Issues:

1. This draft only describes the existing in-the-wild protocol, which has some warts. We need an updated, standards-track protocol.

2. IANA registration of the GSSAPI/Kerberos service name is requested. Should IANA registration of the service port also be requested?

3. Should timeouts be more fully specified? Any other issues with error-handling?

4. The original UMICH code has provisions for DSA private keys in addition to RSA. Nobody seems to use DSA. Everyone seems to use RSA. Is it worth specifying how DSA would work?

Changes from Draft -02 to Draft -03:

1. The abstract was expanded.

2. Additional information was provided on traditional UDP size restrictions and their effect on reliability and key sizes in section 3.

3. The updates to the security considerations for digital signature usage were incomplete, and have been rewritten.

4. Information on an optional client version feature (which does not appear to be actually in use) was added to the request ASN.1, and Appendix B, and the title of the appendix changed.

5. As before some minor changes to wording were made for clarity, but are not believed to have changed the meaning.

Changes from Draft -01 to Draft -02:

1. The retry behavior was made slightly less specific.

2. The traditionally used SAN extensions were moved to a new appendix, leaving only the id-pkinit-san as the RECOMMENDED SAN.

3. The absolute prohibition against digital signatures in the Security Considerations section was relaxed since there are legitimate situations where a signature based on the KX509 certificate is still useful. (E.g. integrity protection where the actual signing identity is not important.)

4. Reference to TAGPMA in the abstract was replaced with a reference to its parent, the International Grid Trust Federation, and more detailed informative references were expanded in the Introduction.

5. Assorted other wording changes were made for clarity, but are not believed to have changed the meaning.

Author's Address

Henry B. Hotz
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
US

Phone: +01 818 354-4880
Email: hotz@jpl.nasa.gov