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

M. Short, Ed.
S. Moore
P. Miller
Microsoft Corporation
October 27, 2014

Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) Freshness Extension
draft-short-pkinit-freshness-00

Abstract

This document describes how to extend Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) extension [RFC4556] to exchange an opaque data blob which a KDC can validate to ensure that the client is currently in possession of the private key during a PKInit AS exchange.

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 PKINIT [RFC4556] defines two schemes to use asymmetric cryptography in a Kerberos preauthenticator. One uses Diffie-Hellman key exchange and the other depends on public key encryption. The public key encryption scheme is less commonly used for two reasons:

- Elliptic Curve Cryptography (ECC) Support [RFC5349] only supports Elliptic Curve Diffie-Hellman (ECDH) key agreement.

- Requires certificates with an encryption key which is not deployed on many existing smart cards.

In the Diffie-Hellman exchange, the client uses its private key only to sign the AuthPack specified in Section 3.2.1 of [RFC4556] which is performed before any traffic is sent to the KDC. Thus a client can generate requests with future times in the PKAuthenticator, and then send those requests at the future times. Unless the time is outside the validity period of the client's certificate, the KDC will validate it and return a TGT the client can use without possessing the private key.

As a result, a client performing PKINIT with the Diffie-Hellman key exchange does not prove current possession of the private key being

used for authentication. It proves only prior use of that key. Ensuring that the client has current possession of the private key requires that the signed PKAuthenticator data include information that the client could not have predicted in advance.

1.1. Kerberos message flow using KRB_AS_REQ without pre-authentication

Today some password-based AS exchanges [RFC4120] depend on the client sending a KRB_AS_REQ without pre-authentication to trigger the KDC to provide the Kerberos client with information needed to complete an AS exchange such as the supported encryption types and salt value (see message flow below):

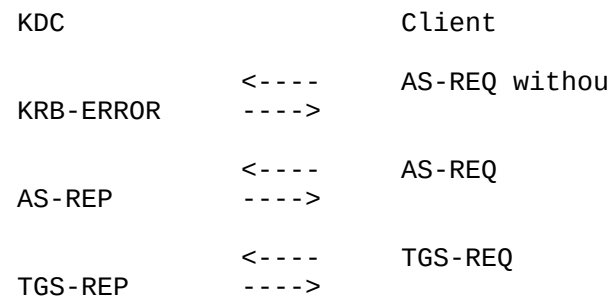

Figure 1

We can use this mechanism in PKInit for KDCs to provide data which the client returns as part of the KRB_AS_REQ to ensure that the PA_PK_AS_REQ [RFC4556] was not pregenerated.

1.2. 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. Message Exchanges

This following summarizes the message flow with extensions to [RFC4120] and [RFC4556] required to support a KDC provided freshness token during the initial request for a ticket:

1. The client generates a KRB_AS_REQ with the OPT-HARDWARE-AUTH option specified in Section 2.9.3 [RFC4120] to the KDC.

2. The KDC generates a KRB_ERROR as specified in Section 3.1.3 of [RFC4120] providing a freshness token.

3. The client receives the error as specified in Section 3.1.4 of [RFC4120] and includes the freshness token as part of the KRB_AS_REQ as specified in [RFC4120] and [RFC4556].

4. The KDC receives and validates the KRB_AS_REQ as specified in Section 3.2.2 [RFC4556] then additionally validates the freshness token.

5. The KDC and client continue as specified in [RFC4120] and [RFC4556].

2.1. Generation of KRB_ERROR Message

The KDC will indicate support by adding to the METHOD-DATA object the PA-PK-AS-KDCTOKEN with padata-type is PA_PK_AS_KDCTOKEN.

2.2. Generation of KRB_AS_REQ Message

After the client receives the KRB-ERROR message, when generating the PKInit AS-REQ it extracts PA-PK-AS-KDCTOKEN as an opaque data blob. When generating the PKAuthenticator, the PA-PK-AS-KDCTOKEN SHALL be added as an opaque blob in the kdcToken field so it becomes part of the signed data in the KRB_AS_REQ.

2.3. Receipt of KRB_AS_REQ Message

After validating the PA_PK_AS_REQ message normally, the KDC will validate the PA-PK-AS-KDCTOKEN in an implementation specific way. If the freshness token is not valid, the KDC MUST return KDC_ERR_PREAUTH_FAILED with PA-PK-AS-KDCTOKEN. Since the freshness tokens are validated by KDCs in the same realm, standardizing the contents of the freshness token is not a concern for interoperability.

3. PreAuthentication Data Types

The following are the new PreAuthentication data types:

Padata and Data Type	Padata-type Value
PA_PK_AS_KDCTOKEN	TBD

4. PA-PK-AS-KDCTOKEN

The PA-PK-AS-KDCTOKEN structure specifies an freshness token. Its structure is defined using ASN.1 notation. The syntax is as follows:

```
PA-PK-AS-KDCTOKEN ::= OCTET STRING
```

5. Extended PKAuthenticator

The PKAuthenticator structure specified in Section 3.2.1 [RFC4556] is extended to include a new kdcToken as follows:

```
PKAuthenticator ::= SEQUENCE {
   cusec        [0] INTEGER (0..999999),
   ctime        [1] KerberosTime,
             -- cusec and ctime are used as in [RFC4120], for
             -- replay prevention.
   nonce        [2] INTEGER (0..4294967295),
             -- Chosen randomly;  this nonce does not need to
             -- match with the nonce in the KDC-REQ-BODY.
   paChecksum   [3] OCTET STRING OPTIONAL,
             -- MUST be present.
             -- Contains the SHA1 checksum, performed over
             -- KDC-REQ-BODY.
   ...,
   kdcToken     [4] PA-PK-AS-KDCTOKEN OPTIONAL,
             -- MUST be present if sent by KDC
   ...
}
```

6. Acknowledgements

Nathan Ide and Magnus Nystrom reviewed the document and provided suggestions for improvements.

7. IANA Considerations

IANA is requested to assign numbers for PA_PK_AS_KDCTOKEN listed in the Kerberos Parameters registry Pre-authentication and Typed Data as follows:

Type	Value	Reference
TBD	PA_PK_AS_KDCTOKEN	[This RFC]

8. Security Considerations

The freshness token SHOULD include either signing or sealing data from the KDC to prevent tampering. Kerberos error messages are not integrity protected unless authenticated using Kerberos FAST [RFC6113].

The freshness token SHOULD include signing, encrypting or sealing data from the KDC to determine authenticity. Even if FAST is required to provide integrity protection, a different KDC would not be able to validate freshness tokens without some kind of shared database.

Since the client treats the KDC provided data blob as opaque, changing the contents will not impact existing clients. Thus extensions to the freshness token do not impact client interoperability.

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.

[RFC5349] Zhu, L., Jaganathan, K., and K. Lauter, "Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 5349, September 2008.

9.2. Informative References

[RFC6113] Hartman, S. and L. Zhu, "A Generalized Framework for Kerberos Pre-Authentication", RFC 6113, April 2011.

Authors' Addresses

Michiko Short (editor)
Microsoft Corporation
USA

Email: michikos@microsoft.com

Seth Moore
Microsoft Corporation
USA

Email: sethmo@microsoft.com

Paul Miller
Microsoft Corporation
USA

Email: paumil@microsoft.com