TOC

Network Working Group	L. Hornquist Astrand
Internet-Draft	Apple, Inc
Intended status: Standards Track	December 05, 2008
Expires: June 8, 2009	

Kerberos ticket extensions
draft-ietf-krb-wg-ticket-extensions-00

Status of this Memo

Abstract

The Kerberos protocol does not allow ticket extensions. This make it harder to deploy features like PKCROSS.

Since the Kerberos protocol did not specified extensibility for the Ticket structure and the current implementations are aware of the contents of tickets, the extension protocol cannot simply extend the Ticket ASN.1 structure. Instead, the extension data needs to be hidden inside the ticket.

This protocol defines two methods to add extend the tickets. The first method requires updated clients and is more in line with the future development of Kerberos. The second way does not require update client. To take advantage of this protocol the server (KDC or application server) need to update a well. The two methods are equivalent and there is a 1-1 mapping between them.

Table of Contents

1. Requirements Notation

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

2. Background

TOC

The ticket and enc-part as defined by [RFC4120] (Neuman, C., Yu, T., Hartman, S., and K. Raeburn, “The Kerberos Network Authentication Service (V5),” July 2005.) is defined as follow:

```
     Ticket          ::= [APPLICATION 1] SEQUENCE {
             tkt-vno         [0] INTEGER (5),
             realm           [1] Realm,
             sname           [2] PrincipalName,
             enc-part        [3] EncryptedData -- EncTicketPart
     }

     EncryptedData   ::= SEQUENCE {
             etype   [0] Int32 -- EncryptionType --,
             kvno    [1] UInt32 OPTIONAL,
             cipher  [2] OCTET STRING -- ciphertext
     }
```

The reason that the ticket can't be extended is that Kerberos clients parses the returned ticket and any additions field will not be preserved.

3. Extending the ticket

TOC

This document describe two methods to extend tickets in Section 3.1 (Update Kerberos 5 ticket) and Section 3.2 (Backward compatible format). The two methods are equivalent and there is a 1-1 mapping between them, copy the fields into the respetive fields. Anyone that creates protocols that uses ticket extentions MUST support the Section 3.2 (Backward compatible format) and SHOULD support both, ie, not depend on the encoding of the Ticket structure itself.

3.1. Update Kerberos 5 ticket

TOC

The first method to extend the ticket is add a new field, ext-data that extends the ticket with an array of type-value ticket extensions.

```
     Ticket ::= [APPLICATION 1] SEQUENCE {
            tkt-vno[0]              Int32,
            realm[1]                Realm,
            sname[2]                PrincipalName,
            enc-part[3]             EncryptedData
            ext-data[4]             SEQUENCE OF TicketExtension OPTIONAL
     }
```

The client signals support by sending the PA-DATA type pa-data-Client-Extensions setting the bit Client-Extensions-support-et-ticket (the zero bit) to 1.
XXX write IANA registration for pa-data-Client-Extensions. Require standard action, private/experimental gets to use their define their own pa data.
If the KDC implement any protocols that uses Ticket extentions, it MUST implement this method. Clients MAY support it. Servers MUST support if they told the KDC they support the extended keys via an administative command.

3.2. Backward compatible format

TOC

The second method is the backward compatible ticket that doesn't change the format of the Ticket structure uses hides the extension data inside the enc-part of the ticket.
It does this by using a special encryption type etype-TBETicket to signal that enc-part.cipher contains the DER-encoded TBETicket structure, instead of an encrypted EncTicketPart.

```
     etype-TBETicket INTEGER ::= 4711 -- TBA XXX --

     krb5int32  ::= INTEGER (-2147483648..2147483647)

     TBETicket ::= SEQUENCE {
             etype           [0] krb5int32 -- EncryptionType --,
             cipher          [1] OCTET STRING
             extensions      [2] SEQUENCE OF TicketExtension OPTIONAL
     }
```

The content of cipher data and encryption type fields is moved inside TBETicket. The kvno field is not moved and have the same mening as before.

If the KDC implement this protocol, it MUST support this method, Clients MAY support it. Servers MUST support if they told the KDC they support the extended keys via an administative command.

4. Ticket extentions

TOC

Ticket extentions are for communicating between the KDC and the service/KDC the ticket is for. Clients and 4th parties can read the data, but should do no attempt to modify, remove or add extentions. The ticket extentions them self is defined as follows:

```
     TicketExtension ::= SEQUENCE {
             te-type [0] krb5int32,
             te-data [1] OCTET STRING
             te-csum [2] Checksum OPTIONAL,
             te-kvno [3] krb5int32 OPTIONAL
     }
```

Negative ticket extension types (te-type) is private extensions and MUST only be used for experimentation or private use.
The te-type field specifies the type of the content in te-data. Unknown te-types MUST be ignored both by the client and the server.
The te-csum field is optional for the type, specified by each ticket extension type. The ticket extension type have to be specified and the key usage number to use for the check sum. The key is usually the session key of the ticket, but doesn't have to be, an extension could specify an new session key used for the ticket. The data that is signed is also specifed specific type.
The (te-kvno) field is to allow changing keys if they keys is some unrelated key.
The KDC MUST NOT use extended ticket in an AS or TGS reply unless it is known that all instances of the service in question support it. In particular, a (local or cross-realm) TGT MUST NOT use extended tickets unless all of the KDCs to which it may be sent are known to support it.
The KDC MAY return extended tickets to servers supporting ticket extensions even if the extended ticket does not contain any extensions.

5. How to request a new assignment for a ticket extension

TOC

When anyone is writing a Internet-draft for which a new assignment for te-type is needed/wanted under the ticket extension, then the proper way to do so is as follows:

```
           EXAMPLE-MODULE DEFINITIONS ::= BEGIN

           krb5-ticket-extension-Name ::= INTEGER nnn
           -- IANA: please assign nnn
           -- RFC-Editor: replace nnn with IANA-assigned
           --               number and remove this note
           END
```

IANA: Don't do note above, its an example, remove this note RFC-Editor: Don't do note above, its an example, remove this note IANA will assign the number as part of the RFC publication process.

When reviewing the document, the reviewer should take sure to check that if te-csum is used, the signing key and key usage is specified. The data that is signed also needs to be specified.

6. Security Considerations

TOC

This document describes how to extend Kerberos tickets to include additional data in the ticket. This does have a security implications since the extension data in the TBETicket is only optionally signed, not encrypted and is not replay protected. It is up to the consumers of this interface to make sure its used safely.

Some of the issues that the extensions need to protect them self from are: MITM downgrade to normal ticket, add or remove extensions, cut and paste extensions between requests, retransmission of requests to a different KDC. The data is sent in clear text, so can should be taken to not send private data.

The ticket extension is mainly to communicate information from the KDC to the server. The information can either be protected by the session key, or the key of the server. If its protected by the session key they both the client and the server can modify the data, and if its protected the servers key is can modified by the server. Any extension using Kerberos extension needs to define what the data is needs protection from.

7. Acknowledgements

TOC

Thanks to Leif Johansson, Kamada Ken'ichi, and Ken Raeburn for reviewing the document and provided suggestions for improvements.

8. IANA Considerations

TOC

There are currently no ticket extensions. Future ticket extensions will be published at:

```
http://www.iana.org/assignments/NNNNNNNN
-- IANA: please name registry, proposal: krb5-ticket-extensions
```

IANA is requested to maintain this registry for future assignments. New assignments can only be made via Specification Required as described in [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.).
IANA will assign the number as part of the RFC publication process.

9. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2434]	Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” BCP 26, RFC 2434, October 1998 (TXT, HTML, XML).
[RFC4120]	Neuman, C., Yu, T., Hartman, S., and K. Raeburn, “The Kerberos Network Authentication Service (V5),” RFC 4120, July 2005 (TXT).

Appendix A. Ticket-extensions ASN.1 Module

TOC

```
KerberosV5-TicketExtensions {
       iso(1) identified-organization(3) dod(6) internet(1)
       security(5) kerberosV5(2) modules(4) ticket-extensions(TBA)
--- XXX who is the registerar for this number ?
} DEFINITIONS EXPLICIT TAGS ::= BEGIN

IMPORTS
        -- as defined in RFC 4120
        Int32, Checksum
                FROM KerberosV5Spec2 { iso(1) identified-organization(3)
                     dod(6) internet(1) security(5) kerberosV5(2)
                     modules(4) krb5spec2(2) }


pa-data-Client-Extensions INTEGER ::= 4710 -- XXX TBA --

PA-DATA-CLIENT-EXTENSIONS ::= BIT STRING

Client-Extensions-support-et-ticket INTEGER ::= 1

Ticket ::= [APPLICATION 1] SEQUENCE {
       tkt-vno[0]              Int32,
       realm[1]                Realm,
       sname[2]                PrincipalName,
       enc-part[3]             EncryptedData
       ext-data[4]             SEQUENCE OF TicketExtension OPTIONAL
}

etype-TBETicket INTEGER ::= 4711 -- XXX TBA --

TBETicket ::= SEQUENCE {
        etype           [0] Int32 -- EncryptionType --,
        cipher          [1] OCTET STRING
        extensions      [2] SEQUENCE OF TicketExtension OPTIONAL
}

TicketExtension ::= SEQUENCE {
        te-type [0] Int32,
        te-data [1] OCTET STRING
        te-csum [2] Checksum
}

END
```

Appendix B. Changes

TOC

RFC-EDITOR: please remove this section.

*Version lha-krb-wg-ticket-extensions-00 - initial version, after review of Leif Johansson, Kamada Ken'ichi

*Version lha-krb-wg-ticket-extensions-01 - comments from Ken Raeburn: experimentation or private use, attack types, asn1. nits.

*Version lha-krb-wg-ticket-extensions-02 - comments from Ken Raeburn: new format for the Ticket PDU message. protocol neg from the client via pa-data. kvno is Ticket.enc-data, added kvno for te-csum field. Clearifed between what parties the messages are for.

*Version ietf-krb-wg-ticket-extensions-00 - make the Backward compatible format default and MUST support. KDC must always support both. Comment from Sam Hartman.

Author's Address

TOC

Love Hornquist Astrand
Apple, Inc
Cupertino
USA
Email: lha@apple.com

Full Copyright Statement

TOC

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.