

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 18, 2009

L. Hornquist Astrand
Apple, Inc
September 14, 2008

Kerberos ticket extensions
draft-lha-krb-wg-ticket-extensions-02

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 18, 2009.

Internet-Draft

Kerberos ticket extensions

September 2008

Abstract

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

Internet-Draft

Kerberos ticket extensions

September 2008

Table of Contents

1.	Requirements Notation	4
2.	Background	5
3.	Extending the ticket	6
3.1.	Update Kerberos 5 ticket	6
3.2.	Backward compatible format	6
4.	Ticket extentions	8
5.	How to request a new assignment for a ticket extension	9
6.	Security Considerations	10
7.	Acknowledgements	11
8.	IANA Considerations	12
9.	Normative References	13
Appendix A.	Ticket-extensions ASN.1 Module	14
Appendix B.	Changes	15
	Author's Address	16
	Intellectual Property and Copyright Statements	17

1. Requirements Notation

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.](#) Background

The ticket and enc-part as defined by [\[RFC4120\]](#) 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

This document describe two methods to extend tickets in [Section 3.1](#) and [Section 3.2](#). The two methods are equivalent and there is a 1-1 mapping between them, copy the fields into the respective fields. Anyone that creates protocols that uses ticket extensions MUST support the [Section 3.1](#) and SHOULD support both, ie, not depend on the encoding of the Ticket structure itself.

[3.1.](#) Update Kerberos 5 ticket

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 and prefer its use.

[3.2.](#) Backward compatible format

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 MAY support this method.

Ticket extensions 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 extensions.

The ticket extensions 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 (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

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.

[6.](#) Security Considerations

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

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

8. IANA Considerations

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](#)].

IANA will assign the number as part of the RFC publication process.

9. Normative References

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

[Appendix A](#). Ticket-extensions ASN.1 Module

```
KerberosV5-TicketExtensions {  
    iso(1) identified-organization(3) dod(6) internet(1)  
    security(5) kerberosV5(2) modules(4) ticket-extensions(TBA)  
--- XXX who is the registrar 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

```


- o Version 00 - initial version, after review of Leif Johansson, Kamada Ken'ichi
- o Version 01 - comments from Ken Raeburn: experimentation or private use, attack types, asn1. nits.
- o Version 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. Clarified between what parties the messages are for.

Author's Address

Love Hornquist Astrand
Apple, Inc
Cupertino
USA

Email: lha@apple.com

Internet-Draft

Kerberos ticket extensions

September 2008

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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.

Hornquist Astrand

Expires March 18, 2009

[Page 17]