NETWORK WORKING GROUP
Internet-Draft
Updates: 4120 (if approved)
Intended status: Standards Track
Expires: April 14, 2007

L. Zhu
P. Leach
K. Jaganathan
Microsoft Corporation

October 11, 2006

Anonymity Support for Kerberos
draft-ietf-krb-wg-anon-02

Status of this Memo

Copyright Notice

Abstract

This document defines extensions to the Kerberos protocol for the Kerberos client to authenticate the Kerberos Key Distribution Center and the Kerberos server, without revealing the client's identity. These extensions can be used to secure communication between the anonymous client and the server.

Table of Contents

1. Introduction

In certain situations, the Kerberos [RFC4120] client may wish to authenticate a server and/or protect communications without revealing its own identity. For example, consider an application which provides read access to a research database, and which permits queries by arbitrary requestors. A client of such a service might wish to authenticate the service, to establish trust in the information received from it, but might not wish to disclose its identity to the service for privacy reasons.

Extensions to [RFC4120] are specified in this document by which a client can authenticate the KDC and request an anonymous ticket. The client can use the anonymous ticket to authenticate the server and protect subsequent client-server communications. These extensions provide Kerberos with functional equivalence to Transport Layer Security (TLS) [RFC4346].

By using the extensions defined in this specification, the client MAY reveal its identity in its initial request to its own KDC, but it can remain anonymous thereafter to KDCs on the cross-realm authentication path, and to the server with which it communicates.

2. 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].

3. Definitions

The anonymous Kerberos realm name is a reserved realm name based on [KRBNAM]. The value is the literal "RESERVED:ANONYMOUS".

The anonymous Kerberos principal name is a reserved Kerberos principal name based on [KRBNAM]. The value of the name-type field is KRB_NT_RESRVED [KRBNAM], and the value of the name-string field is a sequence of two KerberosString components: "RESERVED", "ANONYMOUS".

Note that in this specification, the anonymous principal name and realm are only applicable to the client in Kerberos messages, the server MUST NOT be anonymous in any Kerberos message.

The transited field [RFC4120] of a ticket is an anonymous authentication path if the tr-type field of the TransitedEncoding type [RFC4120] is NO-TRANSITED-INFO and the contents field is an

empty OCTET STRING.

```
NO-TRANSITED-INFO    TBA
```

This means that no information of the authentication path is disclosed.

The anonymous ticket flag is defined as bit TBA (with the first bit being bit 0) in the TicketFlags:

```
TicketFlags     ::= KerberosFlags
  -- anonymous(TBA)
  -- TicketFlags and KerberosFlags are defined in [RFC4120]
```

An anonymous ticket is a ticket that has all of the following properties:

- The cname field [RFC4120] contains the anonymous Kerberos principal name.
- The crealm field [RFC4120] contains either the client's realm name or the anonymous realm name.
- The transited field [RFC4120] can contain either the client's authentication path as described in Section 3.3.3.2 of [RFC4120] or the anonymous authentication path.
- The anonymous ticket contains no information that can reveal the client's identity. However the ticket MAY contain the client realm and the realms on the authentication path, and authorization data that MAY provide information related to the client's identity. For example, an anonymous principal that is only identifiable within a particular group of users can be implemented using authorization data and such authorization data, if included in the anonymous ticket, shall disclose the client's membership of that group.
- The anonymous ticket flag is set.

The request-anonymous KDC option is defined as bit TBA (with the first bit being bit 0) in the KDCOptions:

```
KDCOptions      ::= KerberosFlags
  -- request-anonymous(TBA)
  -- KDCOptions and KerberosFlags are defined in [RFC4120]
```

4. Protocol Description

In order to request an anonymous ticket, the client sets the request-anonymous KDC option in an Authentication Exchange (AS) or Ticket Granting Service (TGS) request [RFC4120]. The client can request an anonymous TGT based on a normal TGT. If the client wishes to authenticate the KDC anonymously, it sets the client name as anonymous in the AS exchange and provides a PA_PK_AS_REQ pre-authentication data [RFC4556] where both the signerInfos field and the certificates field of the SignedData [RFC3852] of PA_PK_AS_REQ are empty. Because the anonymous client does not have an associated asymmetric key pair, the client MUST use the Diffie-Hellman key agreement method by filling in the Diffie-Hellman domain parameters in the clientPublicValue [RFC4556].

If the ticket in the PA-TGS-REQ [RFC4120] of the TGS request is anonymous, or if the client in the AS request is anonymous, the request-anonymous KDC option MUST be set in the request.

Upon receiving the AS request with a PA_PK_AS_REQ from the anonymous client, the KDC skips the checks for the client's signature and the client's public key (such as the verification of the binding between the client's public key and the client name), but performs otherwise-applicable checks, and proceeds as normal according to [RFC4556]. For example, the AS MUST check if the client's Diffie-Hellman domain parameters are acceptable. The Diffie-Hellman key agreement method MUST be used and the reply key is derived according to Section 3.2.3.1 of [RFC4556]. If the clientPublicValue is not present in the request, the KDC MUST return a KRB-ERROR [RFC4120] with the code KDC_ERR_PUBLIC_KEY_ENCRYPTION_NOT_SUPPORTED [RFC4556] and there is no accompanying e-data. The client that made the anonymous request can authenticate the KDC based on the KDC's signature in the reply. If the KDC does not have an asymmetric key pair, it MAY reply anonymously. In which case, both the signerInfos field and the certificates field of the SignedData [RFC3852] of PA_PK_AS_REP in the reply are empty. The server name in an anonymous reply contains the name of the TGS. Upon receipt of an anonymous KDC reply, the client MUST reject the returned ticket if it can not authenticate the KDC otherwise.

The client can use its keys to mutually authenticate with the KDC, and request an anonymous TGT in the AS request. And in that case, the reply key is selected as normal according to Section 3.1.3 of [RFC4120].

For the TGS exchange, the reply key is selected as normal according to Section 3.3.3 of [RFC4120].

When policy allows, the KDC issues an anonymous ticket. Based on local policy, the client realm in the anonymous ticket can be the anonymous realm name or the realm of the KDC. However, in all cases, the client name and the client realm in the EncKDCRepPart of the reply [RFC4120] MUST match with the corresponding client name and the client realm of the anonymous ticket in the reply. The client MUST use the client name and the client realm returned in the EncKDCRepPart in subsequent message exchanges when using the obtained anonymous ticket.

During the TGS request, when propagating authorization data, care MUST be taken by the TGS to ensure that the client confidentiality is not violated. The TGS MUST either fail the request or remove authorization data that may reveal the client's identity. An optional authorization element unknown by the TGS MUST be removed if it can be ignored (such as ones enclosed in the AD-IF-RELEVANT structure). The TGS can only strip critical unknown authorization data if the ticket does not convey any rights such as those conveyed by a KDCIssued authorization data element. If a ticket contains a KDCIssued authorization data element, then no other authorization data elements may be removed if they could serve to limit the rights conveyed by the KDCIssued element. Here is a table of the known authorization-data elements, tagged with whether they interfere with client anonymity and recommendations for how to process them:

ad-type	References	Can Breach Confidentiality?
AD-IF-RELEVANT	RFC4120	Yes, remove if unknown
AD-KDCIssued	RFC4120	Yes, fail the request if unknown
AD-AND-OR	RFC4120	Yes, remove if unknown
AD-MANDATORY-FOR-KDC	RFC4120	Yes, fail the request if unknown

The KDC fills out the transited field of the anonymous ticket in the reply as follows: If the service ticket in a TGS request is an anonymous ticket with a "normal" authentication path, then the authentication path in the reply ticket MUST also contain a "normal" authentication path, the TGS MUST add the name of the previous realm. However, if the service ticket in a TGS request is an anonymous ticket with an anonymous authentication path, then the reply ticket can contain either an anonymous authentication path or a "normal" authentication path, based on local policy of the KDC. Thus a "normal" authentication path in an anonymous ticket can be a partial path, it may not include all the intermediate realms on the authentication path.

The KDC fills out the authtime field of the anonymous ticket in the reply as follows: If the anonymous ticket is returned in an AS exchange, the authtime field of the ticket contains the request time.

If the anonymous ticket is returned in a TGS exchange, the authtime field contains the authtime of the ticket in the PA-TGS-REQ [RFC4120]. An anonymous ticket can be renewed, and the authtime field of a renewed ticket is the authtime in the anonymous ticket on which the renewed ticket was based.

If it is inappropriate to remove an authorization element from the TGS request in order to produce an anonymous ticket, the KDC MUST return an error message with the code KDC_ERR_POLICY [RFC4120].

If the client is anonymous and the KDC does not have a key to encrypt the reply, the KDC MUST return an error message with the code KDC_ERR_NULL_KEY [RFC4120] and there is no accompanying e-data.

If a client requires anonymous communication then the client MUST check to make sure that the ticket in the reply is actually anonymous by checking the presence of the anonymous ticket flag. This is because KDCs ignore unknown KDC options. A KDC that does not understand the request-anonymous KDC option will not return an error, but will instead return a normal ticket.

The subsequent client and server communications then proceed as described in [RFC4120]. No transited policy checking is needed for the anonymous authentication path. However, transited policy checks defined in Section 2.7 of [RFC4120] would apply to an anonymous ticket that contains a "normal" authentication path.

A server accepting an anonymous service ticket may assume that subsequent requests using the same ticket originate from the same client. Requests with different tickets are likely to originate from different clients.

Interoperability and backward-compatibility notes: the KDC is given the task of rejecting a request for an anonymous ticket when the anonymous ticket is not acceptable by the server.

5. GSS-API Implementation Notes

At the GSS-API [RFC2743] level, the use of an anonymous principal by the initiator/client requires the initiator/client to assert the "anonymous" flag when calling GSS_Init_Sec_Context().

GSS-API does not know or define "anonymous credentials", so the (printable) name of the anonymous principal will rarely be used by or relevant for the initiator/client. The printable name is relevant for the acceptor/server when performing an authorization decision based on the name that pops up from GSS_Accept_Sec_Context() upon

successful security context establishment.

A GSS-API initiator MUST carefully check the resulting context attributes from the initial call to GSS_Init_Sec_Context() when requesting anonymity, because (as in the GSS-API tradition and for backwards compatibility) anonymity is just another optional context attribute. It could be that the mechanism doesn't recognize the attribute at all or that anonymity is not available for some other reasons -- and in that case the initiator must NOT send the initial security context token to the acceptor, because it will likely reveal the initiators identity to the acceptor, something that can rarely be "un-done".

GSS-API defines the name_type GSS_C_NT_ANONYMOUS [RFC2743] to represent the anonymous identity. In addition, Section 2.1.1 of [RFC1964] defines the single string representation of a Kerberos principal name with the name_type GSS_KRB5_NT_PRINCIPAL_NAME. For the anonymous principals, the name component within the exportable name as defined in Section 2.1.3 of [RFC1964] MUST signify the realm name according to Section 2.1.1 of [RFC1964]. Note that in this specification only the client/initiator can be anonymous.

Portable initiators are RECOMMENDED to use default credentials whenever possible, and request anonymity only through the input anon_req_flag [RFC2743] to GSS_Init_Sec_Context().

6. Security Considerations

Since KDCs ignore unknown options [RFC4120], a client requiring anonymous communication needs to make sure that the ticket is actually anonymous. This is because a KDC that that does not understand the anonymous option would not return an anonymous ticket.

By using the mechanism defined in this specification, the client does not reveal its identity to the server but its identity may be revealed to the KDC of the server principal (when the server principal is in a different realm than that of the client), and any KDC on the cross-realm authentication path. The Kerberos client MUST verify the ticket being used is indeed anonymous before communicating with the server, otherwise the client's identity may be revealed unintentionally.

In cases where specific server principals must not have access to the client's identity (for example, an anonymous poll service), the KDC can define server principal specific policy that insure any normal service ticket can NEVER be issued to any of these server principals.

If the KDC that issued an anonymous ticket were to maintain records of the association of identities to an anonymous ticket, then someone obtaining such records could breach the anonymity. Additionally, the implementations of most (for now all) KDC's respond to requests at the time that they are received. Traffic analysis on the connection to the KDC will allow an attacker to match client identities to anonymous tickets issued. Because there are plaintext parts of the tickets that are exposed on the wire, such matching by a third party observer is relatively straightforward.

7. Acknowledgements

Clifford Neuman contributed the core notions of this document.

Martin Rex wrote the text for GSS-API considerations.

Nicolas Williams reviewed the GSS-API considerations section and suggested ideas for improvements.

Sam Hartman and Nicolas Williams were great champions of this work.

In addition, the following individuals made significant contributions: Jeffery Altman, Tom Yu, Chaskiel M Grundman, Love Hoernquist Aestrand, and Jeffery Hutzelman.

8. IANA Considerations

Section 3 defines the anonymous Kerberos name and the anonymous Kerberos realm based on [KRBNAM]. The IANA registry for [KRBNAM] need to be updated to add references to this document.

9. Normative References

[KRBNAM] Zhu, L., "Additonal Kerberos Naming Contraints", draft-ietf-krb-wg-naming, work in progress.

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996.

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

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)",

RFC 3852, July 2004.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

Authors' Addresses

Larry Zhu
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: lzhu@microsoft.com

Paul Leach
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: paulle@microsoft.com

Karthik Jaganathan
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

Email: karthikj@microsoft.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).