Kerberos Working Group
Internet-Draft
Intended status: Experimental
Expires: November 4, 2011

A. Perez-Mendez
R. Marin-Lopez
F. Pereniguez-Garcia
G. Lopez-Millan
University of Murcia
May 3, 2011

GSS-API pre-authentication for Kerberos
draft-gss-preauth-00

Abstract

This document describes a pre-authentication mechanism for Kerberos based on the Generic Security Service Application Program Interface (GSS-API), which allows a Key Distribution Center (KDC) to authenticate clients by using any GSS mechanism.

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 November 4, 2011.

Copyright Notice

Table of Contents

1. Introduction

The GSS-API (Generic Security Service Application Programming Interface) [RFC2743] provides a generic toolset of functions that allow applications to establish security contexts in order to protect their communications through security services such as authentication, confidentiality and integrity protection. Thatks to the GSS-API, applications remain independent from the specific underlying mechanism used to establish the context and provide security.

Kerberos [RFC4120] defines a process called pre-authentication. This feature is intended to avoid the security risk of providing tickets encrypted with the user's long-term key to attackers. The execution of a pre-authentication mechanism may require the exchange of several KRB_AS_REQ/KRB_ERROR messages before the KDC delivers the TGT requested by the client within a KRB_AS_REP. These messages transport authentication information by means of pre-authentication elements.

There exists a variety of pre-authentication mechanisms, like PKINIT [RFC4556] and encrypted time-stamp [RFC4120]. Furthermore, [I-D.ietf-krb-wg-preauth-framework] provides a generic framework for Kerberos pre-authentication, which aims to describe the features that a pre-authentication mechanism may provide (e.g. mutual authentication, replace reply key, etc.). Additionally, in order to simplify the definition of new pre-authentication mechanisms, it defines a mechanism called FAST (Flexible Authentication Secure Tunneling), which provides a generic and secure transport for pre-authentication elements. More specifically, FAST establishes a secure tunnel providing confidentiality and integrity protection between the client and the KDC prior to the exchange of any specific pre-authentication data. Within this tunnel, different pre-authentication methods can be executed. This inner mechanism is called a FAST factor. It is important to note that FAST factors cannot usually be used outside the FAST pre-authentication method

since they assume the underlying security layer provided by FAST.

The aim of this draft is to define a new pre-authentication mechanism, following the recommendations of [I-D.ietf-krb-wg-preauth-framework], that relies on the GSS-API security services to pre-authenticate clients. This pre-authentication mechanism will allow the KDC to authenticate clients making use of any current or future GSS mechanism, as long as they satisfy the minimum security requirements described in this specification. The Kerberos client will play the role of the GSS initiator, while the Authentication Server (AS) in the KDC will play the role of the GSS acceptor.

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. Definition of the Kerberos GSS padata

To establish the security context, the GSS-API defines the exchange of GSS tokens between the initiator and the acceptor. These tokens, which contain mechanism-specific information, are completely opaque to the application. However, how these tokens are transported between the initiator and the responder depends on the specific application. Since GSS-API is defined as independent of the underlying communications service, its use does not require to implement any specific security feature for the transport. For instance, tokens could just be sent by means of plain UDP datagrams. For this reason, security and ordered delivery of information must be implemented by each specific GSS mechanism (if required).

Therefore, GSS tokens are the atomic piece of information from the application point of view when using GSS-API, which require a proper transport between the initiator (Kerberos client) and the acceptor (AS). In particular, the proposed GSS-based pre-authentication mechanism defines a new pre-authentication element (hereafter padata) called PA-GSS-TOKEN to transport a generic GSS token from the Kerberos client to the AS and vice-versa. The ASN.1 specification for this padata is:

PA-GSS-TOKEN To be defined (TBD)

PA-GSS-TOKEN ::= OCTECT STRING --value of the GSS token

3. GSS Pre-authentication Operation

3.1. Generation of GSS preauth requests

The Kerberos client (initiator) starts by calling to the GSS_Init_sec_context function. In the first call to this function, the client provides GSS_C_NO_CTX as the value of the context_handle and NULL as the input_token, given that no context has been initiated yet. When using multi round-trip GSS mechanisms, in subsequent calls to this routine the client will use both, the context_handle value obtained after the first call, and the input_token received from the KDC.

The GSS_Init_sec_context returns a context_handle, an output_token and a status value. If the obtained status is GSS_S_COMPLETE, the generated token contains the neccesary information to establish the context and, therefore, no further tokens are expected from the KDC to complete the authentication process. On the contrary, if the obtained status is GSS_S_CONTINUE_NEEDED, the KDC is expected to provide a token in order to continue with the context establishment process. In both cases, the Kerberos client includes the obtained output_token into a new PA-GSS-TOKEN padata and sends it to the KDC through a KRB_AS_REQ message.

3.2. Processing of GSS preauth requests

When the KDC (GSS acceptor) receives a KRB_AS_REQ message containing a PA-GSS-TOKEN, but a PA-FX-COOKIE (see Section 6) is not included, the KDC assumes that this is the first message of a context establishment, and thus GSS_C_NO_CTX is used as context_handle to invoke the GSS_Accept_sec_context routine. Conversely, if a PA-FX-COOKIE is included, the KDC assumes that this message is part an ongoing authentication and the value of the PA-FX-COOKIE is used as the context_handle value. In both cases, after receiving the message, the KDC calls to the GSS_Accept_sec_context function, using

the adequate context_handle value and using the received token in the PA-GSS-TOKEN padata as input_token.

Once the execution of the GSS_Accept_sec_context function is completed, the KDC obtains a context_handle, an output_token (optional) that MUST be sent to the initiator in order to continue with the authentication process, and a status value. If the obtained status is GSS_S_COMPLETE, the client is considered authenticated and the value of the output_token may be NULL. If the status is GSS_S_CONTINUE_NEEDED, further information is required to complete the process.

3.3. Generation of GSS preauth responses

Once the KDC has processed the input_token provided by the client (as described in Section 3.2, two main different situations may occur depending on the status value. If the client is successfully authenticated (GSS_S_COMPLETE), the KDC will reply to the client with a KRB_AS_REP message. This message will transport the final output_token (if generated) in a PA-GSS-TOKEN padata type. Additionally, there are three alternatives to encrypt the enc-part field of the KRB_AS_REP message. The first one is to make use of the client's password as described in the standard Kerberos. A second option is to strengthen this key by using keying material from the GSS context (more details are provide in Section 5). The final option is to employ a key cryptographically independent from the

user's password which could be generated by using the keying material from the GSS context. Section 5 provides further details regarding these two last options.

On the contrary, if further data is required to complete the establishment process (GSS_S_CONTINUE_NEEDED), the KDC will reply to the client with a KDC_ERR_MORE_PREAUTH_DATA_REQUIRED error message [I-D.ietf-krb-wg-preauth-framework]. In the e-data field of the message, the KDC will include two padata types: a PA-FX-COOKIE containing the context_handle of this context (Section 6), and a PA-GSS-TOKEN containing the obtained output_token.

3.4. Processing of GSS preauth responses

When the client receives a KDC_ERR_MORE_PREAUTH_DATA_REQUIRED error,

it extracts the token from the PA-GSS-TOKEN element and invokes to the GSS_Init_sec_context function, as described in section Section 3.1. The received PA-FX-COOKIE is treated as an opaque element, which is simply copied and included into the following KRB_AS_REQ message without further processing.

On the other hand, when the client receives a KRB_AS_REP, the context establishment has finalized successfully. If the KRB_AS_REP message contains a PA-GSS-TOKEN padata type, the client invokes the GSS_Init_sec_context function using the transported input_token. Note that, to be consistent, this call MUST return GSS_S_COMPLETE and not generate any output_token, since the KDC does not expect further data from the client. Similarly, if the client does not expect any data from the KDC (it obtained a GSS_S_COMPLETE status value on the last call) and the KDC provides an input_token, an unexpected situation occurrs and the context establishment must be aborted.

If the context establishment is completed correctly, the client must use the same process followed by the KDC (Section 3.3).

4. Data in the KDC_ERR_PREAUTH_REQUIRED

When the KDC sends a KDC_ERR_PREAUTH_REQUIRED error to the client, it includes a sequence of padata, each corresponding to an acceptable pre-authentication method. Optionally, these padatas contain data valuable for the client to configure the selected mechanism. The data to be included in the padata for this message is described in this section.

TBD. (For example, list of the OIDs of the GSS mechanisms supported by the KDC)

5. Supported pre-authentication facilities

The pre-authentication framework [I-D.ietf-krb-wg-preauth-framework] defines a set of facilities that the pre-authentication mechanisms may provide. Specifically, the GSS pre-authentication mechanism proposed in this draft may provide the following facilities:

o Client-authentication facility. The GSS pre-authentication

mechanism authenticates the client based on GSS-API calls. At the end of the GSS context establishment process, the client is authenticated against the KDC by means of the specific GSS mechanism credentials.

- Strengthening-reply-key facility. After a successful authentication, client and KDC may strengthen the reply key (the key used to encrypt the enc-part field of the KRB_AS_REP message) by adding additional keying material to it. This additional keying material can be obtained by means of calls to the GSS_Pseudo_random [RFC4401] function, although the standard GSS_getMIC function could be used if the former is not available for the specific GSS mechanism.

- Replacing-reply-key facility. Similarly to the strengthening facility, client and KDC may decide to completely replace the reply key used to encrypt the KRB_AS_REP by a new one that is cryptographically independent from the client's password stored in client password on the Kerberos users database. To generate this keying material, the same GSS-API functions used for the previous facility would be used.

- KDC-authentication facility. This facility is also provided, as an optional feature, since the GSS-API allows the initiator of the security context to request mutual authentication during the establishment process. If the mutual_req_flag is indicated in the GSS_Init_sec_context call, the acceptor (KDC) must be authenticated by the initiator (client) before the context is established.

The selection of the facilities that the GSS pre-authentication mechanism will provide, and how will they be negotiated with the client is still under discussion.

6. Managing states for the KDC

The Kerberos standard [RFC4120] defines the KDC as a stateless entity. This means that, if the GSS mechanism requires more than one round-trip, the client must provide enough data to the KDC in the

following interactions to allow recovering the complete state of the

ongoing authentication. This is specially relevant when the client switches from one KDC to different one (within the same realm) during a pre-authentication process. This second KDC must be able to continue with the process in a seamless way. In [I-D.ietf-krb-wg-preauth-framework], the PA-FX-COOKIE pre-authentication element is defined to transport opaque state information from the KDC to the client. This state information is included by the client in the following KRB_AS_REQ message as-is, without further processing. When the KDC receives the PA-FX-COOKIE padata, it tries to recover the state and, if successful, continue with the authentication process.

PA-FX-COOKIE 133

The GSS-API manages the so-called security contexts. They represent the whole context of an authentication, including all the state and relevant data of the ongoing security context. Every GSS-API function requires an input parameter (called context_handle) which identifies the specific context over which they are applied. The application obtains a value for this context_handle after the first call to the GSS_Init_sec_context function (when acting as GSS initiator) or GSS_Accept_sec_context function (when acting as GSS acceptor), which is used in subsequent calls regarding this security context. Hence, it seems reasonable that this is the value that must be transported in the PA-FX-COOKIE padata type as it allows the KDC to recover the complete state of an ongoing context establishment proccess.

7. Security Considerations

Protection of Request/Responses with FAST, restriction on GSS mechanism, etc. TBD.

8. IANA Considerations

This document has no actions for IANA.

9. Normative References

[I-D.ietf-krb-wg-preauth-framework]
Hartman, S. and L. Zhu, "A Generalized Framework for Kerberos Pre-Authentication", draft-ietf-krb-wg-preauth-framework-17 (work in progress), June 2010.

[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.

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

[RFC4401] Williams, N., "A Pseudo-Random Function (PRF) API Extension for the Generic Security Service Application Program Interface (GSS-API)", RFC 4401, February 2006.

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

Authors' Addresses

Alejandro Perez-Mendez (Ed.)
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 46 44
Email: alex@um.es

Rafa Marin-Lopez
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 85 01
Email: rafa@um.es

Fernando Pereniguez-Garcia
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 78 82
Email: pereniguez@um.es

Gabriel Lopez-Millan
University of Murcia
Campus de Espinardo S/N, Faculty of Computer Science
Murcia, 30100
Spain

Phone: +34 868 88 85 04
Email: gabilm@um.es