Workgroup: Kerberos Working Group
Internet-Draft:
draft-perez-krb-wg-gss-preauth-03
Published: September 2021
Intended Status: Experimental
Expires: 27 March 2022
Authors: A. Perez-Mendez R. Marin-Lopez
Jisc University of Murcia
F. Pereniguez-Garcia G. Lopez-Millan
University Defense Center University of Murcia
L. Howard-Bentata
PADL Software Pty Ltd

GSS-API pre-authentication for Kerberos

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 a 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 https://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 5 March 2022.

Copyright Notice

Table of Contents

1. Introduction

The Generic Security Service Application Programming Interface (GSS-API) [RFC2743] provides a framework for authentication and message protection services through a common programming interface, allowing applications to remain agnostic from the selected mechanism.

Kerberos [RFC4120] is an authentication service based on the Needham-Schroeder symmetric key protocol. It includes a facility called pre-authentication designed to ensure clients prove knowledge of their long-term key before the Key Distribution Center (KDC) issues a ticket. Typical pre-authentication mechanisms include encrypted timestamp [RFC4120] and public key certificates [RFC4556]. Pre-authentication data in these messages provides a typed hole for exchanging information used to authenticate the client.

[RFC6113] specifies a framework for pre-authentication in Kerberos, describing the features such a pre-authentication mechanism may provide such as authenticating the client and/or KDC and strengthening or replacing the reply key in the AS-REP. FAST (Flexible Authentication Secure Tunneling) provides a generic and secure transport for pre-authentication elements prior to the exchange of any pre-authentication data. The inner pre-

authentication mechanism is called a FAST factor. FAST factors can generally not be used outside FAST as they assume the underlying security layer provided by FAST.

This document defines a new pre-authentication method that relies on GSS-API security services to pre-authenticate Kerberos clients. This method allows the KDC to authenticate clients using any current or future GSS-API mechanism, as long as they satisfy the minimum security requirements described in this specification. The Kerberos client assumes the role of the GSS-API initiator, and the Authentication Service (AS) the role of the GSS-API acceptor. It may be used as a FAST factor or without FAST.

This work was originally motivated by the desire to allow Kerberos to use the protocols defined in [RFC7055] to authenticate federated users with EAP.

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 [RFC2119].

2. Prerequisites

2.1. Cookie Support

KDCs which support GSS-API pre-authentication with mechanisms that require more than one round-trip to establish a security context MUST have a secure mechanism for retaining state between AS-REQs. For stateless KDC implementations, this will typically be a digest of the initial KDC-REQ-BODY concatenated with a GSS_Export_sec_context() token, encrypted in a key known only to the KDC and protected from replay attacks (see Section 5.2 of [RFC6113]). The format of the PA-FX-COOKIE is implementation defined.

Clients that support GSS-API pre-authentication with mechanisms that require more than one round-trip MUST echo the received PA-FX-COOKIE in the next AS-REQ (within a given conversation).

2.2. More Pre-Authentication Data Required

Both KDCs and clients which implement GSS-API pre-authentication MUST support the use of KDC_ERR_MORE_PREAUTH_DATA_REQUIRED, as decribed in Section 5.2 of [RFC6113].

2.3. Support for Exporting Partially Established Contexts

KDC implementations that use exported context tokens to maintain state will call GSS_Export_sec_context() and GSS_Import_sec_context() on partially established acceptor contexts. This may require modifications to the mechanism implementation, as [RFC2743] only requires these functions succeed on fully established contexts.

2.4. Processing of Channel Bindings in Single Round-Trip

The client's KDC request is bound to the GSS-API context establishment through the use of channel bindings. GSS-API mechanisms that require more than one round-trip do not expose at which point in the exchange the channel bindings are validated, and assume they are constant for all context establishment calls. In this specification, the channel bindings contain the encoded client request body, which may vary for each round-trip if a fresh nonce is used on each request.

To accommodate this, and to avoid re-encoding the request body without the nonce, this specification imposes the additional requirement that the GSS-API mechanism processes channel bindings in a single round-trip within the pre-authentication conversation.

3. Definition of the GSS padata

The GSS-API defines an exchange of opaque tokens between the initiator (client) and acceptor (service) in order to authenticate each party. GSS-API does not define the transport over which these tokens are carried. This specification defines a Kerberos pre-authentication type, PA-GSS, which carries a GSS-API context token from the Kerberos client to the AS and vice versa.

```
PA-GSS          633
-- output_token from GSS_Init_sec_context()
-- or GSS_Accept_sec_context()
```

4. GSS-API Pre-authentication Operation

4.1. Kerberos client (GSS-API initiator)

The Kerberos client begins by calling GSS_Init_sec_context() with the desired credential handle and the target name of the TGS, including the instance and realm. If the underlying mechanism supports Kerberos names, the TGS name MUST be imported as a GSS_KRB5_NT_PRINCIPAL_NAME; otherwise, it SHALL be imported as a GSS_C_NT_HOSTBASED_SERVICE with "krbtgt" as the "service" element and the TGS realm as the "hostname" element (see [RFC2743] Section 4.1).

In the first call to GSS_Init_sec_context(), input_context_handle is GSS_C_NO_CONTEXT and input_token is empty. In subsequent calls the client uses the context_handle value obtained after the first call, and the input_token received from the KDC. The mutual_req_flag MUST be set.

In order to bind the GSS-API and Kerberos message exchanges, the DER-encoded KDC-REQ-BODY from the AS-REQ is passed as channel binding application data. As the nonce may differ between requests (see [RFC6113] Section 5.4.3), this requires the GSS-API mechanism to process the channel binding information in a single round-trip. To avoid this potential interoperability issue, clients MAY use a single nonce for all messages in a conversation once GSS-API pre-authentication has commenced.

If GSS_Init_sec_context() returns GSS_S_CONTINUE_NEEDED, the output_token is sent to the KDC in the PA-GSS pre-authentication data and the client expects either a KRB-ERROR containing another context token, or an AS-REP optionally containing a final context token.

Once GSS_Init_sec_context() returns GSS_S_COMPLETE, the context is ready for use. The AS-REP is decrypted using the reply key (see Section 6) and the Kerberos client name MAY be replaced by the AS-REP cname (see Section 7). The client MUST fail if the mutual_state flag is not set when fully established, unless the KDC was authenticated by some other means such as a FAST armor.

The response received from the KDC must agree with the expected status from GSS_Init_sec_context(). It is a state violation to receive an AS-REP from the KDC when the initiator still has additional tokens to send to the KDC (GSS_S_CONTINUE_NEEDED), or conversely to receive KDC_ERR_MORE_PREAUTH_DATA_REQUIRED if the context from the initiator's perspective was already open (GSS_S_COMPLETE).

When receiving a KDC_ERR_MORE_PREAUTH_DATA_REQUIRED error from the KDC, an PA-FX-COOKIE from the KDC MUST be present and copied into the subsequent AS-REQ.

4.2. KDC (GSS-API acceptor)

When the KDC receives an AS-REQ message containing PA-GSS pre-authentication data, it first looks for an PA-FX-COOKIE and if present retrieves the context handle associated with the cookie, typically by passing the context token from the decrypted cookie to GSS_Import_sec_context(). The absence of an PA-FX-COOKIE indicates a new conversation and the client sending an initial context token.

The KDC SHALL associate the KDC-REQ-BODY of the initial request with the pre-authentication conversation. On subsequent requests, the KDC MUST abort the conversation and return an error if the KDC-REQ-BODY differs from the initial request. The nonce is excluded from this comparison. This extends the protection afforded by the channel binding to all requests in the conversation, not just the request where the mechanism validated the channel bindings. (No specific implementation is required, but one approach would be for the KDC to include a digest of the KDC-REQ-BODY with the nonce set to zero in the PA-FX-COOKIE contents.)

If no PA-GSS pre-authentication data is present, the KDC cannot continue with GSS-API pre-authentication and will continue with other pre-authentication methods or return an error as determined by local policy. If PA-GSS pre-authentication data is present but empty, the KDC SHALL return a KDC_ERR_PREAUTH_FAILED error. Otherwise, GSS_Accept_sec_context() is called with the acceptor credential handle, the token provided in the PA-GSS pre-authentication data, and channel binding application data containing the DER-encoded KDC-REQ-BODY.

If GSS_Accept_sec_context() returns GSS_S_CONTINUE_NEEDED, the KDC returns a KDC_ERR_MORE_PREAUTH_DATA_REQUIRED error with the output token included as PA-GSS pre-authentication data. The acceptor state is encoded, typically by calling GSS_Export_sec_context(), and the encrypted result is placed in an PA-FX-COOKIE.

If GSS_Accept_sec_context() returns GSS_S_COMPLETE, the context is ready for use and an AS-REP is returned using the reply key specified in Section 6. Otherwise, an appropriate error such as KDC_ERR_PREAUTH_FAILED is returned to the client and the conversation is aborted. If the mechanism emitted an error token on failure, it SHOULD be returned to the client.

If the GSS-API mechanism requires an odd number of messages to establish a security context, the KDC MUST include an empty GSS-PA pre-authentication data in the last message of a successful conversation.

5. Indication of Supported Mechanisms

When the KDC sends a KDC_ERR_PREAUTH_REQUIRED error to the client, it MAY include a pre-authentication data element indicating the set of supported mechanisms. The pre-authentication data comprises of a SPNEGO server initiated initial context token as defined in [MS-SPNG] 3.2.5.2, containing the list of mechanisms supported by the acceptor. Context state is discarded and as such the first PA-GSS from the client is always an InitialContextToken ([RFC2743] Section 3.1).

6. Reply Key Derivation

The GSS-API pre-authentication mechanism proposed in this draft provides the Replace Reply Key facility [RFC6113].

After authentication is complete, the client and KDC replace the AS-REP reply key with the output of calling GSS_Pseudo_random() [RFC4401] with the following parameters:

context The initiator or acceptor context handle

prf_key GSS_C_PRF_KEY_FULL

prf_in KRB-GSS || 0x00 || AS-REQ nonce

desired_output_len The length in bytes of original reply key

The nonce is the nonce of the final AS-REQ in the conversation, and is encoded as the little-endian binary representation of 4 bytes. The new reply key has the same key type as the original key. If FAST is used, the new reply key SHOULD be strengthened by including a strengthen key in the KrbFastResponse.

7. Naming

This specification permits Kerberos clients to authenticate without knowing how the KDC will map their GSS-API initiator name to a Kerberos principal. In such cases the client SHALL set the value of the cname field in the AS-REQ to the well-known [RFC6111] value WELLKNOWN/FEDERATED, replacing it after a successful conversation with the client name returned in the AS-REP.

When the initiator knows the Kerberos client name it wishes to authenticate as, and the mechanism supports Kerberos names, the name MUST be imported using the GSS_KRB5_NT_PRINCIPAL_NAME name type. Otherwise, GSS_C_NT_USER_NAME SHOULD be used when importing NT-PRINCIPAL names in the local realm, or NT-ENTERPRISE [RFC6806] names. GSS_C_NT_HOSTBASED_SERVICE SHOULD be used when importing NT-SRV-HOST or NT-SRV-INST names with a single instance.

This specification does not mandate a specific mapping of GSS-API initiator names to Kerberos principal names. KDCs MAY use the NT-ENTERPRISE principal name type to avoid conflating any domain- or realm-like components of initiator names with Kerberos realms.

The KDC MAY include an AD-GSS-COMPOSITE-NAME authorization data element, containing name attribute information. Its value is the exp_composite_name octet string resulting from a successful call to GSS_Export_name_composite() [RFC6680]. It SHOULD be enclosed in a AD-IF-RELEVANT container. The format of composite name tokens is

implementation dependent; services that cannot parse the name token MUST fail if the authorization data element was not enclosed in AD-IF-RELEVANT.

8. Anonymous Authentication

If the client wishes to authenticate anonymously using GSS-API pre-authentication, it MUST specify both the request-anonymous flag in the AS-REQ and anon_req_flag in the call to GSS_Init_sec_context(). If GSS_Accept_sec_context() set anon_state and returned an initiator name of type GSS_C_NT_ANONYMOUS, the KDC MUST map the user to the well-known anonymous PKINIT principal and realm defined in [RFC8062].

If GSS_Accept_sec_context() set anon_state but did not return an initiator name of type GSS_C_NT_ANONYMOUS, then the KDC MUST return the well-known anonymous principal but it MAY include the realm of the initiator.

9. Security Considerations

The client SHOULD use FAST armor to protect the pre-authentication conversation.

The KDC MUST maintain confidentiality and integrity of the PA-FX-COOKIE contents, typically by encrypting it using a key known only to itself. Cookie values SHOULD be protected from replay attacks by limiting their validity period and binding their contents to the client name in the AS-REQ.

The establishment of a GSS-API security context is bound to the client's AS-REQ through the inclusion of the encoded KDC-REQ-BODY as channel bindings (see Section 4.1), and the nonce as input to the key derivation function (see Section 6). By asserting the KDC-REQ-BODY does not change during the conversation (nonce notwithstanding), the channel bindings protect all request bodies in the conversation.

The KDC MAY wish to restrict the set of GSS-API mechanisms it will accept requests from. When using SPNEGO [RFC4178] with GSS-API pre-authentication, the client should take care not to select a mechanism with weaker security properties than a different non-GSS-API pre-authentication type that could have been used.

If mutual_state is false after GSS_Init_sec_context() completes, the client MUST ensure that the KDC was authenticated by some other means.

10. IANA Considerations

Assign PA-GSS value in Pre-authentication and Typed Data, Kerberos Parameters registry (preference for 633).

The ad-type number 633 (TBD) is assigned for AD-GSS-COMPOSITE-NAME, updating the table in Section 7.5.4 of [RFC4120].

11. Normative References

[MS-SPNG] "Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extension", <https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-spng/f377a379-c24f-4a0f-a3eb-0d835389e28a>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, DOI 10.17487/RFC2743, January 2000, <https://www.rfc-editor.org/info/rfc2743>.

[RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, DOI 10.17487/RFC4120, July 2005, <https://www.rfc-editor.org/info/rfc4120>.

[RFC4178] Zhu, L., Leach, P., Jaganathan, K., and W. Ingersoll, "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", RFC 4178, DOI 10.17487/RFC4178, October 2005, <https://www.rfc-editor.org/info/rfc4178>.

[RFC4401] Williams, N., "A Pseudo-Random Function (PRF) API Extension for the Generic Security Service Application Program Interface (GSS-API)", RFC 4401, DOI 10.17487/RFC4401, February 2006, <https://www.rfc-editor.org/info/rfc4401>.

[RFC4556] Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, DOI 10.17487/RFC4556, June 2006, <https://www.rfc-editor.org/info/rfc4556>.

[RFC6111] Zhu, L., "Additional Kerberos Naming Constraints", RFC 6111, DOI 10.17487/RFC6111, April 2011, <https://www.rfc-editor.org/info/rfc6111>.

[RFC6113] Hartman, S. and L. Zhu, "A Generalized Framework for Kerberos Pre-Authentication", RFC 6113, DOI 10.17487/RFC6113, April 2011, <https://www.rfc-editor.org/info/rfc6113>.

[RFC6680] Williams, N., Johansson, L., Hartman, S., and S. Josefsson, "Generic Security Service Application Programming Interface (GSS-API) Naming Extensions", RFC 6680, DOI 10.17487/RFC6680, August 2012, <https://www.rfc-editor.org/info/rfc6680>.

[RFC6806] Hartman, S., Ed., Raeburn, K., and L. Zhu, "Kerberos Principal Name Canonicalization and Cross-Realm Referrals", RFC 6806, DOI 10.17487/RFC6806, November 2012, <https://www.rfc-editor.org/info/rfc6806>.

[RFC7055] Hartman, S., Ed. and J. Howlett, "A GSS-API Mechanism for the Extensible Authentication Protocol", RFC 7055, DOI 10.17487/RFC7055, December 2013, <https://www.rfc-editor.org/info/rfc7055>.

[RFC8062] Zhu, L., Leach, P., Hartman, S., and S. Emery, Ed., "Anonymity Support for Kerberos", RFC 8062, DOI 10.17487/RFC8062, February 2017, <https://www.rfc-editor.org/info/rfc8062>.

Authors' Addresses

Alejandro Perez-Mendez
Jisc
4 Portwall Lane
Bristol
BS1 6NB
United Kingdom

Email: alex.perez-mendez@jisc.ac.uk

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

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

Fernando Pereniguez-Garcia
University Defense Center
Spanish Air Force Academy

30720 San Javier Murcia
Spain

Phone: +34 968 18 99 46
Email: fernando.pereniguez@cud.upct.es

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

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

Luke Howard-Bentata
PADL Software Pty Ltd
PO Box 59
Central Park Victoria 3145
Australia

Email: lukeh@padl.com