Internet Engineering Task Force (IETF)
Request for Comments: 6339
Category: Standards Track
ISSN: 2070-1721

S. Josefsson
SJD AB
L. Hornquist Astrand
Apple, Inc.

August 2011

Context Token Encapsulate/Decapsulate and OID Comparison Functions for the Generic Security Service Application Program Interface (GSS-API)

Abstract

This document describes three abstract Generic Security Service Application Program Interface (GSS-API) interfaces used to encapsulate/decapsulate context tokens and compare OIDs. This document also specifies C bindings for the abstract interfaces.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6339.

Copyright Notice

Table of Contents

1. Introduction

The Generic Security Service Application Program Interface (GSS-API) [RFC2743] is a framework that provides security services to applications using a variety of authentication mechanisms. There are widely implemented C bindings [RFC2744] for the abstract interface.

For initial context tokens, a mechanism-independent token format may be used (see Section 3.1 of [RFC2743]). Some protocols, e.g., Simple Authentication and Security Layer (SASL) GS2 [RFC5801], need the ability to add and remove this token header, which contains some ASN.1 tags, a length, and the mechanism OID to and from context tokens. This document adds two GSS-API interfaces (GSS_Encapsulate_token and GSS_Decapsulate_token) so that GSS-API libraries can provide this functionality.

Being able to compare OIDs is useful, for example, when validating that a negotiated mechanism matches the requested one. This document adds one GSS-API interface (GSS_OID_equal) for this purpose.

Text from this specification can be used as implementation documentation, and for this reason, Sections 3, 4, 5, 6, and 8 should be considered code components.

2. Conventions Used in This Document

The document uses terms from, and is structured in a similar way as, [RFC2743] and [RFC2744]. The normative reference to [RFC5587] is for the C types "gss_const_buffer_t" and "gss_const_OID"; nothing else from that document is required to implement this document.

3. GSS_Encapsulate_token Call

Inputs:

- input_token OCTET STRING -- buffer with token data to encapsulate

- token_oid OBJECT IDENTIFIER -- object identifier of mechanism for the token

Outputs:

- major_status INTEGER

- output_token OCTET STRING -- Encapsulated token data; caller must release with GSS_Release_buffer()

Return major_status codes:

- GSS_S_COMPLETE indicates that completion was successful and that output parameters hold correct information.

- GSS_S_FAILURE indicates that encapsulation failed for reasons unspecified at the GSS-API level.

GSS_Encapsulate_token() is used to add the mechanism-independent token header to GSS-API context token data.

3.1. gss_encapsulate_token

```
OM_uint32 gss_encapsulate_token (
  gss_const_buffer_t input_token,
  gss_const_OID token_oid,
  gss_buffer_t output_token)
```

Purpose:

Add the mechanism-independent token header to GSS-API context token data.

Parameters:

input_token   buffer, opaque, read
                       Buffer with GSS-API context token data.

token_oid   Object ID, read
                       Object identifier of token.

output_token buffer, opaque, modify
Encapsulated token data; caller must release with gss_release_buffer().

Function values: GSS status codes

GSS_S_COMPLETE Indicates that completion was successful and that output parameters hold correct information.

GSS_S_FAILURE Indicates that encapsulation failed for reasons unspecified at the GSS-API level.

4. GSS_Decapsulate_token Call

Inputs:

- input_token OCTET STRING -- buffer with token to decapsulate
- token_oid OBJECT IDENTIFIER -- expected object identifier of token

Outputs:

- major_status INTEGER
- output_token OCTET STRING -- Decapsulated token data; caller must release with GSS_Release_buffer()

Return major_status codes:

- GSS_S_COMPLETE indicates that completion was successful and that output parameters hold correct information.
- GSS_S_DEFECTIVE_TOKEN means that the token failed consistency checks (e.g., OID mismatch or ASN.1 DER length errors).
- GSS_S_FAILURE indicates that decapsulation failed for reasons unspecified at the GSS-API level.

GSS_Decapsulate_token() is used to remove the mechanism-independent token header from an initial GSS-API context token.

4.1. gss_decapsulate_token

```
OM_uint32
gss_decapsulate_token (
  gss_const_buffer_t input_token,
  gss_const_OID token_oid,
  gss_buffer_t output_token)
```

Purpose:

Remove the mechanism-independent token header from an initial GSS-API context token.

Parameters:

input_token buffer, opaque, read
Buffer with GSS-API context token.

token_oid Object ID, read
Expected object identifier of token.

output_token buffer, opaque, modify
Decapsulated token data; caller must release with gss_release_buffer().

Function values: GSS status codes

GSS_S_COMPLETE Indicates that completion was successful and that output parameters hold correct information.

GSS_S_DEFECTIVE_TOKEN Means that the token failed consistency checks (e.g., OID mismatch or ASN.1 DER length errors).

GSS_S_FAILURE Indicates that decapsulation failed for reasons unspecified at the GSS-API level.

5. GSS_OID_equal Call

Inputs:

- first_oid OBJECT IDENTIFIER -- first object identifier to compare
- second_oid OBJECT IDENTIFIER -- second object identifier to compare

Return codes:

- non-0 when neither OID is GSS_C_NO_OID and the two OIDs are equal.
- 0 when the two OIDs are not identical or either OID is equal to GSS_C_NO_OID.

GSS_OID_equal() is used to add compare two OIDs for equality. The value GSS_C_NO_OID will not match any OID, including GSS_C_NO_OID itself.

5.1. gss_oid_equal

```
extern int
gss_oid_equal (
  gss_const_OID first_oid,
  gss_const_OID second_oid
)
```

Purpose:

Compare two OIDs for equality. The value GSS_C_NO_OID will not match any OID, including GSS_C_NO_OID itself.

Parameters:

first_oid	Object ID, read First object identifier to compare.
second_oid	Object ID, read Second object identifier to compare.
Function values:	GSS status codes
non-0	Neither OID is GSS_C_NO_OID, and the two OIDs are equal.
0	The two OIDs are not identical, or either OID is equal to GSS_C_NO_OID.

6. Test Vector

For the GSS_Encapsulate_token function, if the "input_token" buffer is the 3-byte octet sequence "foo" and the "token_oid" OID is 1.2.840.113554.1.2.2, which encoded corresponds to the 9-byte-long octet sequence (using C notation) "\x2a\x86\x48\x86\xf7\x12\x01\x02\x02", the output should be the 16-byte-long octet sequence (again in C notation) "\x60\x0e\x06\x09\x2a\x86\x48\x86\xf7\x12\x01\x02\x02\x66\x6f\x6f". These values may also be used to test the GSS_Decapsulate_token interface.

7. Acknowledgements

Greg Hudson pointed out the 'const' problem with the C bindings in earlier versions of this document, and Luke Howard suggested to resolve it by using the [RFC5587] types. Stephen Farrell suggested several editorial improvements and the security consideration regarding absent security features of the encapsulation function. Chris Lonvick suggested some improvements.

8. Security Considerations

The security considerations of the base GSS-API specification ([RFC2743]) and the base C bindings ([RFC2744]) are inherited.

Encapsulation of data does not provide any kind of integrity or confidentiality.

Implementations need to treat input as potentially untrustworthy for purposes of dereferencing memory objects to avoid security vulnerabilities. In particular, ASN.1 DER length fields are a common source of mistakes.

9. References

9.1. Normative References

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

[RFC2744] Wray, J., "Generic Security Service API Version 2 : C-bindings", RFC 2744, January 2000.

[RFC5587] Williams, N., "Extended Generic Security Service Mechanism Inquiry APIs", RFC 5587, July 2009.

9.2. Informative Reference

[RFC5801] Josefsson, S. and N. Williams, "Using Generic Security Service Application Program Interface (GSS-API) Mechanisms in Simple Authentication and Security Layer (SASL): The GS2 Mechanism Family", RFC 5801, July 2010.

Authors' Addresses

Simon Josefsson
SJD AB
Hagagatan 24
Stockholm 113 47
SE

EMail: simon@josefsson.org
URI: http://josefsson.org/

Love Hornquist Astrand
Apple, Inc.

EMail: lha@apple.com