Internet Engineering Task Force (IETF)
Request for Comments: 6542
Updates: 4121
Category: Standards Track
ISSN: 2070-1721

S. Emery
Oracle

March 2012

Kerberos Version 5
Generic Security Service Application Program Interface (GSS-API) Channel Binding Hash Agility

Abstract

Currently, channel bindings are implemented using an MD5 hash in the Kerberos Version 5 Generic Security Service Application Programming Interface (GSS-API) mechanism (RFC 4121). This document updates RFC 4121 to allow channel bindings using algorithms negotiated based on Kerberos crypto framework as defined in RFC 3961. In addition, because this update makes use of the last extensible field in the Kerberos client-server exchange message, extensions are defined to allow future protocol extensions.

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/rfc6542.

Copyright Notice

Table of Contents

1. Introduction

With the recently discovered weaknesses in the MD5 hash algorithm (see [RFC6151]), there is a need to use stronger hash algorithms. The Kerberos Version 5 Generic Security Service Application Programming Interface (GSS-API) mechanism [RFC4121] uses MD5 to calculate channel binding verifiers. This document specifies an update to the mechanism that allows it to create channel binding information based on negotiated algorithms. This will allow deploying new algorithms incrementally without breaking interoperability with older implementations when new attacks arise in the future.

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].

The term "little-endian order" is used for brevity to refer to the least-significant-octet-first encoding, while the term "big-endian order" is used for the most-significant-octet-first encoding.

3. Channel Binding Hash Agility

When generating a channel binding verifier, Bnd, a hash is computed from the channel binding fields. Initiators MUST populate the Bnd field in order to maintain interoperability with existing acceptors. In addition, initiators MUST populate the extension field (Exts) defined below.

3.1. Structure of the Exts Field

The 0x8003 GSS checksum has the same structure described in [RFC4121] except that the Exts field is now defined; the entire structure of the 0x8003 checksum, including the now defined Exts field, follows:

Octet	Name	Description
0..3	Lgth	Number of octets in Bnd field, represented in little-endian order; currently contains hex value 10 00 00 00 (16).
4..19	Bnd	Channel binding information, as described in Section 4.1.1.2 of [RFC4121].
20..23	Flags	Four-octet context-establishment flags in little-endian order as described in Section 4.1.1.1 of [RFC4121].
24..25	DlgOpt	The delegation option identifier (=1) in little-endian order [optional]. This field and the next two fields are present if and only if GSS_C_DELEG_FLAG is set as described in Section 4.1.1.1 of [RFC4121].
26..27	Dlgth	The length of the Deleg field in little-endian order [optional].
28..(n-1)	Deleg	KRB_CRED message (n = Dlgth + 28) [optional].
n..last	Exts	Extensions.

where Exts is the concatenation of zero, one, or more individual extensions, each of which consists of the following, in order:

type -- big-endian-order unsigned integer, 32 bits, which contains the type of extension
length -- big-endian-order unsigned integer, 32 bits, which contains the length, in octets, of the extension data encoded as an array of octets immediately following this field
data -- octet string of extension information

If multiple extensions are present, then there MUST be at most one instance of a given extension type.

3.2. The Channel Binding Extension

When channel binding is used, the Exts MUST include the following extension:

data-type 0x00000000

data-value

The output obtained by applying the Kerberos V get_mic operation [RFC3961] with key usage number 43 to the channel binding data as described in [RFC4121], Section 4.1.1.2 (using get_mic instead of MD5). The key used is the sub-session key from the authenticator, if it is present; otherwise, the key used is the session key from the ticket. The get_mic algorithm is chosen as the "required checksum mechanism" for the encryption type of the key used.

Initiators that are unwilling to use an MD5 hash of the channel bindings MUST set the Bnd field to sixteen octets of hex value FF.

4. Security Considerations

With this mechanism, initiators get no indication as to whether the acceptors check or ignore channel bindings.

It is up to the application whether or not to enforce the use of channel bindings. [RFC5056] and [RFC5554] give guidance for application developers on channel binding usage.

5. IANA Considerations

IANA has created a new top-level registry titled "Kerberos V GSS-API Mechanism Parameters," separate from the existing Kerberos parameters registry. Within this registry, IANA has created a sub-registry of "Kerberos V GSS-API Mechanism Extension Types" with four-field entries (Type Number, Type Name, Description, and Reference) and, initially, a single registration: 0x00000000, "Channel Binding MIC," "Extension for the verifier of the channel bindings," [RFC6542].

Using the guidelines for allocation as described in [RFC5226], type number assignments are as follows:

0x00000000 - 0x000003FF IETF Review

0x00000400 - 0xFFFFF3FF Specification Required

0xFFFFF400 - 0xFFFFFFFF Private Use

6. Acknowledgments

The author would like to thank Larry Zhu, Nicolas Williams, Sam Hartman, Jeffrey Hutzelman, and Simon Josefsson for their help in reviewing and providing valuable feedback on this document.

7. References

7.1. Normative References

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

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, February 2005.

[RFC4121] Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

7.2. Informative References

[RFC5056] Williams, N., "On the Use of Channel Bindings to Secure Channels", RFC 5056, November 2007.

[RFC5554] Williams, N., "Clarifications and Extensions to the Generic Security Service Application Program Interface (GSS-API) for the Use of Channel Bindings", RFC 5554, May 2009.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, March 2011.

Author's Address

Shawn Emery
Oracle
500 Eldorado Blvd, Building 1
Broomfield, CO 80021
USA

EMail: shawn.emery@oracle.com