A PRF for the Kerberos V GSS-API Mechanism
draft-ietf-kitten-krb5-gssapi-prf-03.txt

Status of this Memo

Copyright Notice

Abstract

   This document defines the Pseudo-Random Function (PRF) for the
   Kerberos V mechanism for the Generic Security Service Application
   Programming Interface (GSS-API), based on the PRF defined for the
   Kerberos V cryptographic framework, for keying application protocols
   given an established Kerberos V GSS-API security context.

Table of Contents

[1](#). Introduction

   This document specifies the Kerberos V GSS-API mechanism's pseudo-
   random function corresponding to [GSS-PRF]. The function is a "PRF+"
   style construction.

[1.1](#) 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].

[2](#). Kerberos V GSS Mechanism PRF

   The GSS-API PRF [GSS-PRF] function for the Kerberos V mechanism
   [RFC1964] shall be the output of a PRF+ function based on the
   encryption type's PRF function keyed with the negotiated session key
   of the security context corresponding to the 'prf_key' input
   parameter of GSS_Pseudo_random().

   This PRF+ MUST be keyed with the key indicated by the 'prf_key' input
   parameter as follows:

   o GSS_C_PRF_KEY_FULL -- use the sub-session key asserted by the
      acceptor, if any, or the sub-session asserted by the initiator, if
      any, or the Ticket's session key

   o GSS_C_PRF_KEY_PARTIAL -- use the sub-session key asserted by the
      initiator, if any, or the Ticket's session key

   The PRF+ function is a simple counter-based extension of the Kerberos
   V pseudo-random function [RFC3961] for the encryption type of the
   security context's keys:

         PRF+(K, L, S) = truncate(L, T1 || T2 || .. || Tn)

         Tn = pseudo-random(K, n || S)

   where '||' is the concatenation operator, 'n' is encoded as a network
   byte order 32-bit unsigned binary number, truncate(L, S) truncates
   the input octet string S to length L, and pseudo-random() is the
   Kerberos V pseudo-random function [RFC3961].

   The maximum output size of the Kerberos V mechanism's GSS-API PRF
   then is, necessarily, 2^{32} times the output size of the pseudo-
   random() function for the encryption type of the given key.

   When the input size is longer than 2^{14} octets as per [GSS-PRF] and

exceeds an implementation's resources then the mechanism MUST return
GSS_S_FAILURE and GSS_KRB5_S_KG_INPUT_TOO_LONG as the minor status
code.

3. IANA Considerations

This document has no IANA considerations currently. If and when a
relevant IANA registry of GSS-API symbols and constants is created
then the GSS_KRB5_S_KG_INPUT_TOO_LONG minor status code should be
added to such a registry.

4. Security Considerations

Kerberos V encryption types' PRF functions use a key derived from
contexts' session keys and should preserve the forward security
properties of the mechanisms' key exchanges.

Legacy Kerberos V encryption types may be weak, particularly the
single-DES encryption types.

See also [GSS-PRF] for generic security considerations of
GSS_Pseudo_random().

See also [RFC3961] for generic security considerations of the
Kerberos V cryptographic framework.

Care should be taken not to exceed the useful lifetime of an
established security context's session key's useful lifetime as
implementations are not required to prevent overuse of the
GSS_Pseudo_random() function. This can effectively be achieved by
limiting the number of GSS_Pseudo_random() calls to, say, a handful
of calls per-security context.

Use of Ticket session keys, rather than sub-session keys, when
initiators and acceptors fail to assert sub-session keys, is
dangerous as ticket reuse can lead to key reuse, therefore initiators
should assert sub-session keys always, and acceptors should assert
sub-session keys at least when initiators fail to do so..

The computational cost of computing this PRF+ may vary depending on
the Kerberos V encryption types being used, but generally the
computation of this PRF+ gets more expensive as the input and output
octet string lengths grow (note that the use of a counter in the PRF+
construction allows for parallelization). This means that if an
application can be tricked into providing very large input octet
strings and requesting very long output octet strings then that may
constitute a denial of service attack on the application; therefore
applications SHOULD place appropriate limits on the size of any input

octet strings received from their peers without integrity protection.

[5](). Normative References

[CFX]   Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos
            Version 5 GSS-API Mechanism: Version 2".

[GSS-PRF]   Williams, N., "A PRF API extension for the GSS-API".

[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.

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

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

Author's Address

   Nicolas Williams
   Sun Microsystems
   5300 Riata Trace Ct
   Austin, TX 78727
   US

   Email: Nicolas.Williams@sun.com

Intellectual Property Statement

   The IETF takes no position regarding the validity or scope of any
   Intellectual Property Rights or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; nor does it represent that it has
   made any independent effort to identify any such rights. Information
   on the procedures with respect to rights in RFC documents can be
   found in BCP 78 and BCP 79.

   Copies of IPR disclosures made to the IETF Secretariat and any
   assurances of licenses to be made available, or the result of an
   attempt made to obtain a general license or permission for the use of
   such proprietary rights by implementers or users of this
   specification can be obtained from the IETF on-line IPR repository at
   http://www.ietf.org/ipr.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights that may cover technology that may be required to implement
   this standard. Please address the information to the IETF at
   ietf-ipr@ietf.org.

Disclaimer of Validity

   This document and the information contained herein are provided on an
   "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS
   OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET
   ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED,
   INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE
   INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
   WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.