Network Working Group
Internet-Draft
Intended status: Experimental
Expires: July 3, 2008

J. Pezeshki
E. Ertekin
R. Jasani
C. Christou
Booz Allen Hamilton

December 31, 2007

IKEv2 Extensions to Support Robust Header Compression over IPsec (RoHCoIPsec)

draft-ietf-rohc-ikev2-extensions-hcoipsec-05

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 3, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

When using Robust Header Compression (RoHC [ROHC]) in conjunction with IPsec [IPSEC] (i.e. [RoHCOIPSEC]) a mechanism is needed to negotiate RoHC configuration parameters between end-points prior to operation. Internet Key Exchange (IKE) is a mechanism which can be leveraged to handle these negotiations. This document specifies

extensions to Internet Key Exchange (IKEv2 [IKEV2]) that will allow RoHC and its associated configuration parameters to be negotiated for IPsec security associations (SAs).

Table of Contents

1. Introduction

Increased packet header overhead due to IPsec protection can result in inefficient utilization of bandwidth. Coupling RoHC with IPsec offers an efficient way to transfer protected IP traffic.

For proper RoHCoIPsec [ROHCOIPSEC] operation, RoHC requires configuration parameters to be negotiated between the compressor and decompressor, prior to operation. Current specifications of hop-by-hop RoHC schemes negotiate these parameters through a link-layer protocol such as Point-to-Point Protocol (PPP) (i.e. RoHC over PPP [ROHCPPP]). Similarly, key exchange protocols (e.g. IKEv2) are commonly used to negotiate parameters between IPsec peers before a SA can be established. This document proposes the use of IKEv2 to handle RoHC channel configuration for RoHCoIPsec, and details various extensions to IKEv2 which are intended to provide this functionality.

2. RoHC Channel Negotiation

The initialization of a RoHC session requires the negotiation of a set of configuration parameters (e.g. MAX_CID, etc.). As such, a mechanism must exist for a RoHC enabled device to share a list of supported RoHC parameters with its peer, and for the peer to select the appropriate parameters from this list.

Similarly, negotiable parameters must also be shared between IPsec peers before a SA can be established. To perform this negotiation, a key exchange protocol, IKEv2, is commonly used. IKEv2 is an extensible protocol that negotiates parameters via request/response message pairs (i.e. exchanges).

A set of extensions to IKEv2 can be defined, which will allow for RoHC parameters to be negotiated during the creation and rekeying of Child SAs. This new Notify payload will contain values for the set of RoHC parameters to be negotiated between the two RoHC peers.

2.1. Negotiation of RoHC Channel Parameters

RoHC configuration parameters will be negotiated at either the establishment or rekeying of a Child SA. Specifically, a Notify payload will be used during the IKE_AUTH and CREATE_CHILD_SA exchanges to negotiate the RoHCoIPsec session. The Notify payload sent by the initiator will contain the configuration parameters for the RoHC scheme. Upon receipt of the initiator's request, the responder will either ignore the payload (if it doesn't support RoHC or the proposed parameters) or respond with a Notify payload that contains the accepted RoHC channel parameters. These accepted

parameters are subset of the parameters proposed by the initiator, and the parameters supported by the responder (e.g. if the initiator proposes a MAX_CID value of 15, but the responder only supports a MAX_CID value of 13, the responder will respond with a value of 13, which is supported by both parties). Note that only one Notify payload is used to convey RoHC parameters per exchange. If multiple Notify payloads relaying RoHC parameters are received by the responder, all but the first such Notify payload must be dropped.

A new Notify Message Type value, denoted ROHC_SUPPORTED, will be added to indicate that the Notify payload is conveying RoHC channel parameters. Additionally, several fields of the Notify payload (as defined in [IKEV2]) are set as follows:

Critical (1 bit)
This value is set to zero to indicate that the recipient must skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload.

RESERVED (7 bits)
Must be sent as zero, and must be ignored on receipt.

Protocol ID (1 octet)
Since the RoHC parameters are set at SA creation, and thus do not relate to an existing SA, this field must be set to zero.

SPI Size (1 octet)
This value must be set to zero, since no SPI is applicable (RoHC parameters are set at SA creation, thus the SPI has not been defined).

Notify Message Type (2 octets)
This field must be set to ROHC_SUPPORTED.

RoHC configuration parameters will be communicated via a new Notify message type, denoted ROHC_SUPPORTED. The RoHC configuration parameters will be listed within the Notification Data field of the Notify payload, in the following format:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!            MAX_CID            !              MRRU             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!           MAX_HEADER          !         PROFILE LENGTH        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                           PROFILES...                         ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                     INTEGRITY ALGORITHMS...                   ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Notification Data field

MAX_CID (2 octets)

The MAX_CID field indicates the maximum value of a context identifier. This value must be at least 0 and at most 16383 (The value 0 implies having one context).

Suggested value: 15

Note: The value of LARGE_CIDS will be implicitly determined by this value (i.e. if MAX_CID is <= 15, LARGE_CIDS will be assumed to be 0).

MRRU (2 octets)

The MRRU field indicates the maximum reconstructed reception unit (see [ROHC], section 5.1.1).

Suggested value: 0

The MRRU value is used in conjunction with the segmentation protocol defined in RoHC. Since RoHCoIPsec will generally be implemented across multiple link-layer "hops", segmentation will not normally be required. In these cases the MRRU value will be set to zero, indicating that no segment headers are allowed on the channel.

MAX_HEADER (2 octets)

The largest header size in octets that may be compressed.

Suggested value: 168 octets

Note: The MAX_HEADER parameter is not used for all RoHC profiles. If none of the RoHC profiles require this field, this value is ignored.

PROFILE LENGTH (2 octets)
The total number of profiles contained within the PROFILES field (note that each RoHC profile is 2-octets in length).

PROFILES
The set of profiles to be enabled for the RoHC process. Profiles are further detailed in [ROHC]. In addition, several common profiles are defined in [ROHCPROF]. These 16-bit profile identifiers are to be sent in network byte order.

INTEGRITY ALGORITHMS
The set of Integrity Algorithms that may be use to ensure the integrity of the decompressed packets (i.e. ensure that the packets are properly decompressed). Each Integrity Algorithm is represented by a 2-octet value that corresponds to the value listed in [IKEV2-PARA] "For Transform Type 3 (Integrity Algorithm)" section.

Note: The length of this field is inferred from the Notify Payload's "Payload Length" field ([IKEV2], Section 3.10).

Note: The key for this Integrity Algorithm is computed using the same method as is used to compute IPsec's Integrity Algorithm key ([IKEV2], Section 2.17).

Note: When a pair of SAs are created (one in each direction), the RoHC channel parameter FEEDBACK_FOR is set implicitly to the other SA of the pair (i.e. the SA pointing in the reverse direction).

3. Security Considerations

The RoHC parameters negotiated via IKEv2 do not add any new vulnerabilities beyond those associated with the normal operation of IKEv2.

4. IANA Considerations

This document defines a new Notify Message (Status Type). Therefore, IANA is requested to allocate one value from the IKEv2 Notify Message registry to indicate ROHC_SUPPORTED. Note that, since this Notify Message is a Status Type, values ranging from 0 to 16383 must not be allocated for ROHC_SUPPORTED.

5. Acknowledgments

The authors would like to thank Mr. Sean O'Keeffe, Mr. James Kohler, and Ms. Linda Noone of the Department of Defense, as well as Mr. Rich Espy of OPnet for their contributions and support in the development of this document. The authors would also like to thank Mr. Tero Kivinen for providing his technical expertise for this document. In addition, the authors would like to thank the following for their numerous reviews and comments to this document:

- o Dr. Stephen Kent
- o Dr. Carsten Bormann
- o Mr. Lars-Erik Jonnson
- o Mr. Pasi Eronen

Finally, the authors would also like to thank Mr. Tom Conkle, Ms. Michele Casey, and Mr. Etzel Brower.

6. Normative References

[ROHC] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RoHCOIPSEC] Ertekin, E., Christou, C., and R. Jasani, "Integration of Robust Header Compression over IPsec Security Associations", work in progress , June 2006.

[IKEV2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[ROHCPPP] Bormann, C., "Robust Header Compression (ROHC) over PPP", RFC 3241, April 2002.

[AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[ROHCPROF]
Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (RoHCv2): Profiles for RTP, UDP, IP, ESP and UDP Lite", www.iana.org/assignments/ROHC-pro-ids , May 2007.

[IKEV2PARA]
"IKEv2 Parameters", http://www.iana.org/assignments/ikev2-parameters , November 2007.

Authors' Addresses

Jonah Pezeshki
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

Email: pezeshki_jonah@bah.com

Emre Ertekin
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

Email: ertekin_emre@bah.com

Rohan Jasani
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

Email: jasani_rohan@bah.com

Chris Christou
Booz Allen Hamilton
13200 Woodland Park Dr.
Herndon, VA 20171
US

Email: christou_chris@bah.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

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, THE IETF TRUST 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.

Intellectual Property

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.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).