Network Working Group
Request for Comments: 2890
Category: Standards Track

G. Dommety
Cisco Systems

September 2000

Key and Sequence Number Extensions to GRE

Status of this Memo

Copyright Notice

Abstract

GRE (Generic Routing Encapsulation) specifies a protocol for encapsulation of an arbitrary protocol over another arbitrary network layer protocol. This document describes extensions by which two fields, Key and Sequence Number, can be optionally carried in the GRE Header [1].

1. Introduction

The current specification of Generic Routing Encapsulation [1] specifies a protocol for encapsulation of an arbitrary protocol over another arbitrary network layer protocol. This document describes enhancements by which two fields, Key and Sequence Number, can be optionally carried in the GRE Header [1]. The Key field is intended to be used for identifying an individual traffic flow within a tunnel. The Sequence Number field is used to maintain sequence of packets within the GRE Tunnel.

1.1. Specification Language

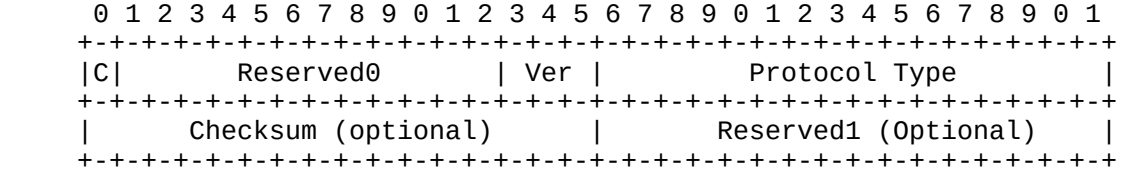
The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

In addition, the following words are used to signify the requirements of the specification.

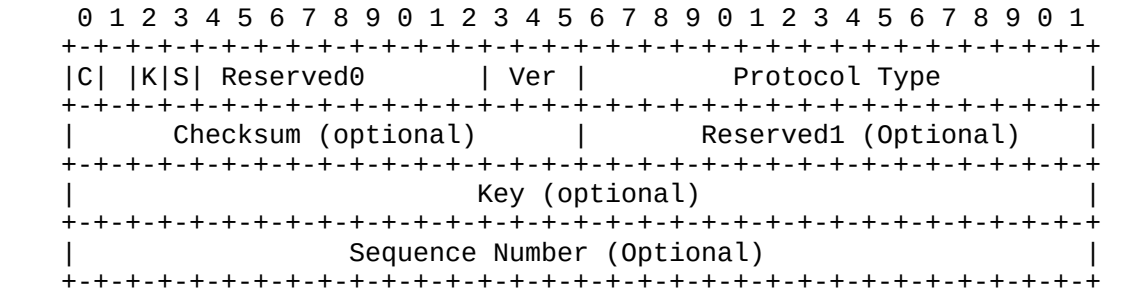
Silently discard
: The implementation discards the datagram without further processing, and without indicating an error to the sender. The implementation SHOULD provide the capability of logging the error, including the contents of the discarded datagram, and SHOULD record the event in a statistics counter.

2. Extensions to GRE Header

The GRE packet header[1] has the following format:



The proposed GRE header will have the following format:



Key Present (bit 2)

If the Key Present bit is set to 1, then it indicates that the Key field is present in the GRE header. Otherwise, the Key field is not present in the GRE header.

Sequence Number Present (bit 3)

If the Sequence Number Present bit is set to 1, then it indicates that the Sequence Number field is present. Otherwise, the Sequence Number field is not present in the GRE header.

The Key and the Sequence Present bits are chosen to be compatible with RFC 1701 [2].

2.1. Key Field (4 octets)

The Key field contains a four octet number which was inserted by the encapsulator. The actual method by which this Key is obtained is beyond the scope of the document. The Key field is intended to be used for identifying an individual traffic flow within a tunnel. For example, packets may need to be routed based on context information not present in the encapsulated data. The Key field provides this context and defines a logical traffic flow between encapsulator and decapsulator. Packets belonging to a traffic flow are encapsulated using the same Key value and the decapsulating tunnel endpoint identifies packets belonging to a traffic flow based on the Key Field value.

2.2. Sequence Number (4 octets)

The Sequence Number field is a four byte field and is inserted by the encapsulator when Sequence Number Present Bit is set. The Sequence Number MUST be used by the receiver to establish the order in which packets have been transmitted from the encapsulator to the receiver. The intended use of the Sequence Field is to provide unreliable but in-order delivery. If the Key present bit (bit 2) is set, the sequence number is specific to the traffic flow identified by the Key field. Note that packets without the sequence bit set can be interleaved with packets with the sequence bit set.

The sequence number value ranges from 0 to (2**32)-1. The first datagram is sent with a sequence number of 0. The sequence number is thus a free running counter represented modulo 2**32. The receiver maintains the sequence number value of the last successfully decapsulated packet. Upon establishment of the GRE tunnel, this value should be set to (2**32)-1.

When the decapsulator receives an out-of sequence packet it SHOULD be silently discarded. A packet is considered an out-of-sequence packet if the sequence number of the received packet is less than or equal to the sequence number of last successfully decapsulated packet. The sequence number of a received message is considered less than or equal to the last successfully received sequence number if its value lies in the range of the last received sequence number and the preceding 2**31-1 values, inclusive.

If the received packet is an in-sequence packet, it is successfully decapsulated. An in-sequence packet is one with a sequence number exactly 1 greater than (modulo 2**32) the last successfully decapsulated packet, or one in which the sequence number field is not present (S bit not set).

If the received packet is neither an in-sequence nor an out-of-sequence packet it indicates a sequence number gap. The receiver may perform a small amount of buffering in an attempt to recover the original sequence of transmitted packets. In this case, the packet may be placed in a buffer sorted by sequence number. If an in-sequence packet is received and successfully decapsulated, the receiver should consult the head of this buffer to see if the next in-sequence packet has already been received. If so, the receiver should decapsulate it as well as the following in-sequence packets that may be present in the buffer. The "last successfully decapsulated sequence number" should then be set to the last packet that was decapsulated from the buffer.

Under no circumstances should a packet wait more that OUTOFORDER_TIMER milliseconds in the buffer. If a packet has been waiting that long, the receiver MUST immediately traverse the buffer in sorted order, decapsulating packets (and ignoring any sequence number gaps) until there are no more packets in the buffer that have been waiting longer than OUTOFORDER_TIMER milliseconds. The "last successfully decapsulated sequence number" should then be set to the last packet so decapsulated.

The receiver may place a limit on the number of packets in any per-flow buffer (Packets with the same Key Field value belong to a flow). If a packet arrives that would cause the receiver to place more than MAX_PERFLOW_BUFFER packets into a given buffer, then the packet at the head of the buffer is immediately decapsulated regardless of its sequence number and the "last successfully decapsulated sequence number" is set to its sequence number. The newly arrived packet may then be placed in the buffer.

Note that the sequence number is used to detect lost packets and/or restore the original sequence of packets (with buffering) that may have been reordered during transport. Use of the sequence number option should be used appropriately; in particular, it is a good idea a avoid using when tunneling protocols that have higher layer in-order delivery mechanisms or are tolerant to out-of-order delivery. If only at certain instances the protocol being carried in the GRE tunnel requires in-sequence delivery, only the corresponding packets encapsulated in a GRE header can be sent with the sequence bit set.

Reordering of out-of sequence packets MAY be performed by the decapsulator for improved performance and tolerance to reordering in the network. A small amount of reordering buffer (MAX_PERFLOW_BUFFER) may help in improving performance when the higher layer employs stateful compression or encryption. Since the state of the stateful compression or encryption is reset by packet loss, it might help the performance to tolerate some small amount of

packet reordering in the network by buffering.

3. Security Considerations

This document describes extensions by which two fields, Key and Sequence Number, can be optionally carried in the GRE (Generic Routing Encapsulation) Header [1]. When using the Sequence number field, it is possible to inject packets with an arbitrary Sequence number and launch a Denial of Service attack. In order to protect against such attacks, IP security protocols [4] MUST be used to protect the GRE header and the tunneled payload. Either ESP (Encapsulating Security Payload) [5] or AH (Authentication Header)[6] MUST be used to protect the GRE header. If ESP is used it protects the IP payload which includes the GRE header. If AH is used the entire packet other than the mutable fields are authenticated. Note that Key field is not involved in any sort or security (despite its name).

4. IANA Considerations

This document does not require any allocations by the IANA and therefore does not have any new IANA considerations.

5. Acknowledgments

This document is derived from the original ideas of the authors of RFC 1701. Kent Leung, Pete McCann, Mark Townsley, David Meyer, Yingchun Xu, Ajoy Singh and many others provided useful discussion. The author would like to thank all the above people.

6. References

[1] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[2] Hanks, S., Li, T, Farinacci, D., and P. Traina, "Generic Routing Encapsulation", RFC 1701, October 1994.

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

[4] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[5] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[6] Kent, S., and R. Atkinson, " IP Authentication Header", RFC 2402, November 1998.

Author's Address

Gopal Dommety
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134

EMail: gdommety@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.