Network Working Group
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: June 19, 2021

M. Jethanandani
Kloud Services
S. Agarwal
Cisco Systems, Inc
A. Mishra
O3b Networks
A. Saxena
Ciena Corporation
A. Dekok
Network RADIUS SARL
December 16, 2020

Secure BFD Sequence Numbers
draft-ietf-bfd-secure-sequence-numbers-07

Abstract

This document describes a security enhancement for the sequence number used in BFD control packets. This document updates RFC 5880.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 19, 2021.

Copyright Notice

Table of Contents

1. Introduction

BFD [RFC5880] section 6.7 describes the use of monotonically incrementing 32-bit sequence numbers for use in authentication of BFD packets. While this method protects against simple replay attacks, the monotonically incrementing sequence numbers are predictable and vulnerable to more complex attack vectors. This document proposes the use of non-monotonically-incrementing sequence numbers in the BFD authentication section to enhance the security of BFD sessions. Specifically, the document presents a method to generate pseudo-random sequence numbers on the frame by algorithmically hashing monotonically increasing sequence numbers. Since the monotonically increasing sequence number does not appear on the wire, it is difficult for a third party to launch a replay attack.

2. Requirements Language

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 RFC 2119 [RFC2119].

3. Theory of operation

Instead of inserting a monotonically, sometimes occasionally, increasing sequence number in BFD control packets, the ciphertext result from a symmetric key algorithm operation (Symmetric-key algorithms require both the sender and the recipient of a message to have the same shared secret key) is inserted. The result is computed, using a shared key, on the sequence number. That

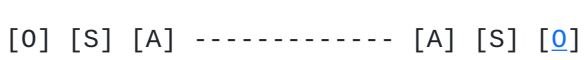
ciphertext result is then inserted into the sequence number field of the packet. In case of BFD Authentication [I-D.ietf-bfd-optimizing-authentication], the sequence number used in computing an authenticated packet would be this new computed ciphertext. Even though the BFD Authentication [I-D.ietf-bfd-optimizing-authentication] sequence number is independent of this enhancement, it would benefit by using the computed ciphertext.

As currently defined in BFD [RFC5880], a BFD packet with authentication will undergo the following steps, where:

[O]: original RFC 5880 packet with monotonically increasing sequence number

[S]: pseudo random sequence number

[A]: Authentication



This document proposes that for enhanced security in sequence number encoding, the sender would identify a symmetric key algorithm that would create a 32 bit ciphertext. The symmetric key is provisioned securely on the sender and receiver of the BFD session. The mechanism of provisioning such a key is outside the scope of this document. This key SHOULD be different from the symmetric key used to to authenticate the packet. Instead of sending the sequence number, the sender encrypts the sequence number using it as input to the symmetric algorithm to produce the ciphertext, which is then inserted in place of the sequence number.

Upon receiving the BFD Control packet, the receiver decrypts the ciphertext using the same provisioned shared key to produce the received sequence number. It compares the received sequence number against the expected sequence number. The mechanism used for comparing is an implementation detail (implementations may pre-calculate the expected sequence number, or decrypt the received sequence number before comparing against expected value). To tolerate dropped frames, the receiver MUST compare the received sequence number against the current expected sequence number (previous received sequence number + 1) and N subsequent expected sequence numbers (where N is greater than or equal to the detect multiplier). Note: The first sequence number can be obtained using the same logic as used in determining Local Discriminator value for the session or by using a random number.

K: symmetric key

S: sequence number

S': encrypted sequence number OR ciphertext result

O: original RFC 5880 packet with monotonically increasing sequence number

f(S, K) = S', where f is a symmetric encryption algorithm

f(S', K) = S, where f is a symmetric decryption algorithm

The above diagram describes how the sender encrypts and receiver decrypts the sequence number. The sender starts by taking the monotonically increasing (but non linear) sequence number and encrypting it using a symmetric encryption algorithm. The resulting ciphertext replaces the sequence number. As per BFD [RFC5880], it then calculates the hash for the entire packet and appends the hash value to the end of the packet, before transmitting it.

The receiver hashes the entire packet as part of receiver authentication. On successful authentication, it decrypts the ciphertext with the same key used to encrypt it, in order to obtain the original sequence number. If it is greater than the previously received monotonically increasing sequence number, then the receiver knows it's a valid sequence number.

4. Impact of using a hash

Under this proposal, every packet's sequence number is encoded in ciphertext. Therefore, there is some impact on the system and its performance while encryption/decryption. As security measures go, this enhancement greatly increases the security of the packet with or without authentication of the entire packet.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

In a symmetric key algorithm, the key is shared between the two systems. Distribution of this key to all the systems at the same time can be quite a cumbersome task. BFD sessions running a fast rate will require these keys to be refreshed often, which poses a further challenge. Therefore, it is difficult to change the keys during the operation of a BFD session without affecting the stability of the BFD session. Therefore, it is recommended to administratively disable the BFD session before changing the keys. If the keys are not changed frequently, an attacker can try to guess the key to launch a replay attack.

This method allows the BFD end-points to detect a malicious packet (the decrypted sequence number will not be in sequence). The behavior of the session, when such a packet is detected, is based on the implementation. A flood of such malicious packets may cause a BFD session to be operationally down.

The symmetric algorithm and key size will determine the difficulty for an attacker to decipher the key from the transmitted BFD frames. The sequential nature of the payload (sequence numbers) simplifies the decoding of the key. It is, therefore, recommended to use longer keys or more secure symmetric algorithms.

7. Acknowledgements

The authors would like to thank Jeff Hass and Reshad Rahman for their reviews of and suggestions for the document.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

8.2. Informative References

[I-D.ietf-bfd-optimizing-authentication]
Jethanandani, M., Mishra, A., Saxena, A., and M. Bhatia, "Optimizing BFD Authentication", draft-ietf-bfd-optimizing-authentication-11 (work in progress), July 2020.

Authors' Addresses

Mahesh Jethanandani
Kloud Services

Email: mjethanandani@gmail.com

Sonal Agarwal
Cisco Systems, Inc
170 W. Tasman Drive
San Jose, CA 95070
USA

Email: agarwaso@cisco.com
URI: www.cisco.com

Ashesh Mishra
O3b Networks

Email: mishra.ashesh@gmail.com

Ankur Saxena
Ciena Corporation
3939 North First Street
San Jose, CA 95134
USA

Email: ankurpsaxena@gmail.com

Alan DeKok
Network RADIUS SARL
100 Centrepointe Drive #200
Ottowa, ON K2G 6B1
Canada

Email: aland@freeradius.org