Internet Engineering Task Force (IETF)
Request for Comments: 7492
Category: Informational
ISSN: 2070-1721

M. Bhatia
Ionos Networks
D. Zhang
Huawei
M. Jethanandani
Ciena Corporation

March 2015

Analysis of Bidirectional Forwarding Detection (BFD) Security According to the Keying and Authentication for Routing Protocols (KARP) Design Guidelines

Abstract

This document analyzes the Bidirectional Forwarding Detection (BFD) protocol according to the guidelines set forth in Section 4.2 of RFC 6518, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines".

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see 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/rfc7492.

Table of Contents

1. Introduction

This document performs a gap analysis of the current state of Bidirectional Forwarding Detection [RFC5880] according to the requirements of KARP Design Guidelines [RFC6518]. Previously, the OPSEC working group has provided an analysis of cryptographic issues with BFD in "Issues with Existing Cryptographic Protection Methods for Routing Protocols" [RFC6039].

The existing BFD specifications provide a basic security solution. Key ID is provided so that the key used in securing a packet can be changed on demand. Two cryptographic algorithms (MD5 and SHA-1) are supported for integrity protection of the control packets; the algorithms are both demonstrated to be subject to collision attacks. Routing protocols like "RIPv2 Cryptographic Authentication" [RFC4822], "IS-IS Generic Cryptographic Authentication" [RFC5310], and "OSPFv2 HMAC-SHA Cryptographic Authentication" [RFC5709] have started to use BFD for liveliness checks. Moving the routing

protocols to a stronger algorithm while using a weaker algorithm for BFD would allow the attacker to bring down BFD in order to bring down the routing protocol. BFD therefore needs to match the routing protocols in its strength of algorithm.

While BFD uses a non-decreasing, per-packet sequence number to protect itself from intra-connection replay attacks, it still leaves the protocol vulnerable to the inter-session replay attacks.

2. Requirements to Meet

There are several requirements described in Section 4 of [RFC6862] that BFD, as defined in BFD [RFC5880], does not currently meet:

Replay Protection: BFD provides an incomplete intra-session and no inter-session replay attack protection; this creates significant denial-of-service opportunities.

Strong Algorithms: The cryptographic algorithms adopted for message authentication in BFD are MD5 or SHA-1 based. However, both algorithms are known to be vulnerable to collision attacks. "BFD Generic Cryptographic Authentication" [BFD-CRYPTO] and "Authenticating BFD using HMAC-SHA-2 procedures" [BFD-HMAC] together propose a solution to support Hashed Message Authentication Code (HMAC) with the SHA-2 family of hash functions for BFD.

Preventing DoS Attacks: BFD packets can be sent at millisecond intervals (the protocol uses timers at microsecond intervals). When malicious packets are sent at short intervals, with the authentication bit set, it can cause a DoS attack. There is currently no lightweight mechanism within BFD to address this issue and is one of the reasons BFD authentication is still not widely deployed in the field.

The remainder of this document explains the details of how these requirements fail to be met and proposes mechanisms for addressing them.

3. Current State of Security Methods

BFD [RFC5880] describes five authentication mechanisms for the integrity protection of BFD control packets: Simple Password, Keyed MD5 [RFC1321], Meticulous Keyed MD5, Keyed SHA-1, and Meticulous Keyed SHA-1. In the simple password mechanism, every control packet is associated with a password transported in plain text; attacks eavesdropping the network traffic can easily learn the password and compromise the security of the corresponding BFD session. In the

Keyed MD5 and the Meticulous Keyed MD5 mechanisms, BFD nodes use shared secret keys to generate Keyed MD5 digests for control packets. Similarly, in the Keyed SHA-1 and the Meticulous Keyed SHA-1 mechanisms, BFD nodes use shared secret keys to generate Keyed SHA-1 digests for control packets. Note that in the keyed authentication mechanisms, every BFD control packet is associated with a non-decreasing, 32-bit sequence number to resist replay attacks. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is only required to increase occasionally. However, in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms, the sequence member is required to increase with each successive packet.

Additionally, limited key updating functionality is provided. There is a Key ID in every authenticated BFD control packet indicating the key used to hash the packet. However, there is no mechanism described to provide a smooth key rollover that the BFD routers can use when moving from one key to the other.

The BFD session timers are defined with the granularity of microseconds, and it is common in practice to send BFD packets at millisecond intervals. Since the cryptographic sequence number space is only 32 bits, a sequence number used in a BFD session may reach its maximum value and roll over within a limited period. For instance, if a sequence number is increased by one every 3.3 milliseconds, then it will reach its maximum value in less than 24 weeks. This can result in potential inter-session replay attacks, especially when BFD uses the non-meticulous authentication modes.

Note that when using authentication mechanisms, BFD drops all packets that fall outside the limited range (3 times the Detection Time multiplier). Therefore, when meticulous authentication modes are used, a replayed BFD packet will be rejected if it cannot fit into a relatively short window (3 times the detection interval of the session). This introduces some difficulties for replaying packets. However, in a non-meticulous authentication mode, such windows can be large (as sequence numbers are only increased occasionally), thus making it easier to perform replay attacks .

In a BFD session, each node needs to select a 32-bit discriminator to identify itself. Therefore, a BFD session is identified by two discriminators. If a node randomly selects a new discriminator for a new session and uses authentication mechanisms to secure the control packets, inter-session replay attacks can be mitigated to some extent. However, in existing BFD demultiplexing mechanisms, the discriminators used in a new BFD session may be predictable. In some deployment scenarios, the discriminators of BFD routers may be decided by the destination and source addresses. So, if the sequence number of a BFD router rolls over for some reason (e.g., reboot), the

discriminators used to identify the new session will be identical to the ones used in the previous session. This makes performing a replay attack relatively simple.

BFD allows a mode called the echo mode. Echo packets are not defined in the BFD specification, though they can keep the BFD session up. The format of the echo packet is local to the sending side, and there are no guidelines on the properties of these packets beyond the choice of the source and destination addresses. While the BFD specification recommends applying security mechanisms to prevent spoofing of these packets, there are no guidelines on what type of mechanisms are appropriate.

4. Impacts of BFD Replays

As discussed, BFD cannot meet the requirements of inter-session or intra-session replay protection. This section discusses the impacts of BFD replays.

When cryptographic authentication mechanisms are adopted for BFD, a non-decreasing, 32-bit-long sequence number is used. In the Keyed MD5 and the Keyed SHA-1 mechanisms, the sequence member is not required to increase for every packet. Therefore, an attacker can keep replaying the packets with the latest sequence number until the sequence number is updated. This issue is eliminated in the Meticulous Keyed MD5 and the Meticulous Keyed SHA-1 mechanisms. However, note that a sequence number may reach its maximum and be rolled over in a session. In this case, without the support from a automatic key management mechanism, the BFD session will be vulnerable to replay attacks performed by sending the packets before the roll over of the sequence number. For instance, an attacker can replay a packet with a sequence number that is larger than the current one. If the replayed packet is accepted, the victim will reject the legal packets whose sequence members are less than the one in the replayed packet. Therefore, the attacker can get a good chance to bring down the BFD session. This kind of attack assumes that the attacker has access to the link when the BFD session is on a point-to-point link or can inject packets for a BFD session with multiple hops.

Additionally, the BFD specification allows for the change of authentication state based on the state of a received packet. For instance, according to BFD [RFC5880], if the state of an accepted packet is down, the receiver of the packet needs to transfer its state to down as well. Therefore, a carefully selected replayed packet can cause a serious denial-of-service attack.

BFD does not provide any solution to deal with inter-session replay attacks. If two subsequent BFD sessions adopt an identical discriminator pair and use the same cryptographic key to secure the control packets, it is intuitive to use a malicious authenticated packet (stored from the past session) to perform interconnection replay attacks.

Any security issues in the BFD echo mode will directly affect the BFD protocol and session states, and hence the network stability. For instance, any replay attacks would be indistinguishable from normal forwarding of the tested router. An attack would still cause a faulty link to be believed to be up, but there is little that can be done about it. However, if the echo packets are guessable, it may be possible to spoof from an external source and cause BFD to believe that a one-way link is really bidirectional. As a result, it is important that the echo packets contain random material that is also checked upon reception.

5. Impact of New Authentication Requirements

BFD can be run in software or hardware. Hardware implementations run BFD at a much smaller timeout, typically in the order of few milliseconds. For instance, with a timeout of 3.3 milliseconds, a BFD session is required to send or receive 3 packets every 10 milliseconds. Software implementations typically run with a timeout in hundreds of milliseconds.

Additionally, it is not common to find hardware support for computing the authentication data for the BFD session in hardware or software. In the Keyed MD5 and Keyed SHA-1 implementation where the sequence number does not increase with every packet, software can be used to compute the authentication data. This is true if the time between the increasing sequence number is long enough to compute the data in software. The ability to compute the hash in software is difficult with Meticulous Keyed MD5 and Meticulous Keyed SHA-1 if the time interval between transmits or between receives is small. The computation problem becomes worse if hundred or thousands of sessions require the hash to be recomputed every few milliseconds.

Smaller and cheaper boxes that have to support a few hundred BFD sessions are boxes that also use a slower CPU. The CPU is used for running the entire control plane software in addition to supporting the BFD sessions. As a general rule, no more than 40-45% of the CPU can be dedicated towards supporting BFD. Adding computation of the hash for every BFD session can easily cause the CPU to exceed the 40-45% limit even with a few tens of sessions. On higher-end boxes with faster and more CPU cores, the expectation is that the number of

sessions that need to be supported are in the thousands, but the number of BFD sessions with authentication that CPU can support is still in the hundreds.

Implementors should assess the impact of authenticating BFD sessions on their platform.

6. Considerations for Improvement

This section suggests changes that can be adopted to improve the protection of BFD.

The security risks brought by SHA-1 and MD5 have been well understood. However, when using a stronger digest algorithm, e.g., SHA-2, the imposed computing overhead will seriously affect the performance of BFD implementation. In order to make the trade-off between the strong algorithm requirement and the imposed overhead, Galois Message Authentication Code (GMAC) can be a candidate option. This algorithm is relatively effective and has been supported by IPsec for data origin authentication. More detailed information can be found in "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH" [RFC4543].

There has been some hallway conversation around the idea of using BFD cryptographic authentication only when some data in the BFD payload changes. The other BFD packets can be transmitted and received without authentication enabled. The bulk of the BFD packets that are transmitted and received have no state change associated with them. Limiting authentication to BFD packets that affect a BFD session state allows for more sessions to be supported for authentication. This change can significantly help the routers since they don’t have to compute and verify the authentication digest for the BFD packets coming at the millisecond intervals. This proposal needs some more discussion in the BFD working group and is certainly a direction that BFD could look at.

7. Security Considerations

This document discusses vulnerabilities in the existing BFD protocol and suggests possible mitigations.

In analyzing the improvements for BFD, the ability to repel a replay attack is discussed. For example, increasing the sequence number to a 64-bit value makes the wrap-around time much longer, and a replay attack can be easily prevented.

Mindful of the impact that stronger algorithms can have on the performance of BFD, the document suggests GMAC as a possible candidate for MAC function.

8. References

8.1. Normative References

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992, <http://www.rfc-editor.org/info/rfc1321>.

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

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010, <http://www.rfc-editor.org/info/rfc6039>.

8.2. Informative References

[BFD-CRYPTO] Bhatia, M., Manral, V., Zhang, D., and M. Jethanandani, "BFD Generic Cryptographic Authentication", Work in Progress, draft-ietf-bfd-generic-crypto-auth-06, April 2014.

[BFD-HMAC] Zhang, D., Bhatia, M., Manral, V., and M. Jethanandani, "Authenticating BFD using HMAC-SHA-2 procedures", Work in Progress, draft-ietf-bfd-hmac-sha-05, July 2014.

[RFC4543] McGrew, D. and J. Viega, "The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH", RFC 4543, May 2006, <http://www.rfc-editor.org/info/rfc4543>.

[RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, February 2007, <http://www.rfc-editor.org/info/rfc4822>.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009, <http://www.rfc-editor.org/info/rfc5310>.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009, <http://www.rfc-editor.org/info/rfc5709>.

[RFC6518] Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", RFC 6518, February 2012, <http://www.rfc-editor.org/info/rfc6518>.

[RFC6862] Lebovitz, G., Bhatia, M., and B. Weis, "Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements", RFC 6862, March 2013, <http://www.rfc-editor.org/info/rfc6862>.

Acknowledgements

We would like to thank Alexander Vainshtein for his comments on this document.

Authors' Addresses

Manav Bhatia
Ionos Networks
Bangalore
India

EMail: manav@ionosnetworks.com

Dacheng Zhang
Huawei

EMail: dacheng.zhang@gmail.com

Mahesh Jethanandani
Ciena Corporation
3939 North 1st Street
San Jose, CA 95134
United States

Phone: 408.904.2160
Fax: 408.436.5582
EMail: mjethanandani@gmail.com