Internet Engineering Task Force
Internet-Draft
Updates: 5880 (if approved)
Intended status: Standards Track
Expires: December 28, 2014

N. Akiya
C. Pignataro
D. Ward
Cisco Systems
M. Bhatia
Ionos Networks
P. K. Santosh
Juniper Networks
June 26, 2014

Seamless Bidirectional Forwarding Detection (S-BFD) draft-ietf-bfd-seamless-base-01

Abstract

This document defines a simplified mechanism to use Bidirectional Forwarding Detection (BFD) with large portions of negotiation aspects eliminated, thus providing benefits such as quick provisioning as well as improved control and flexibility to network nodes initiating the path monitoring.

This document updates RFC5880.

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].

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 http://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 December 28, 2014.

Table of Contents

1. Introduction

Bidirectional Forwarding Detection (BFD), [RFC5880] and related documents, has efficiently generalized the failure detection mechanism for multiple protocols and applications. There are some improvements which can be made to better fit existing technologies. There is a possibility of evolving BFD to better fit new technologies. This document focuses on several aspects of BFD in order to further improve efficiency, to expand failure detection coverage and to allow BFD usage for wider scenarios. This document extends BFD to provide solutions to use cases listed in [I-D.ietf-bfd-seamless-use-case].

One key aspect of the mechanism described in this document eliminates the time between a network node wanting to perform a connectivity test and completing the connectivity test. In traditional BFD terms, the initial state changes from DOWN to UP is virtually nonexistent. Removal of this seam (i.e. time delay) in BFD provides applications a smooth and continuous operational experience. Therefore, "Seamless BFD" (S-BFD) has been chosen as the name for this mechanism.

2. Terminology

The reader is expected to be familiar with the BFD, IP and MPLS terminologies and protocol constructs. This section describes several new terminologies introduced by S-BFD.

- S-BFD - Seamless BFD.
- S-BFD packet - a BFD control packet on the well-known S-BFD port.
- Entity - a function on a network node that S-BFD mechanism allows remote network nodes to perform connectivity test to. An entity can be abstract (ex: reachability) or specific (ex: IP addresses, router-IDs, functions).
- SBFDInitiator - an S-BFD session on a network node that performs a connectivity test to a remote entity by sending S-BFD packets.
- SBFDReflector - an S-BFD session on a network node that listens for incoming S-BFD packets to local entities and generates response S-BFD packets.
- Reflector BFD session - synonymous with SBFDReflector.

- S-BFD discriminator - a BFD discriminator allocated for a local entity and is being listened by an SBFDReflector.

- BFD discriminator - a BFD discriminator allocated for an SBFDInitiator.

- Initiator - a network node hosting an SBFDInitiator.

- Responder - a network node hosting an SBFDReflector.

Below figure describes the relationship between S-BFD terminologies.

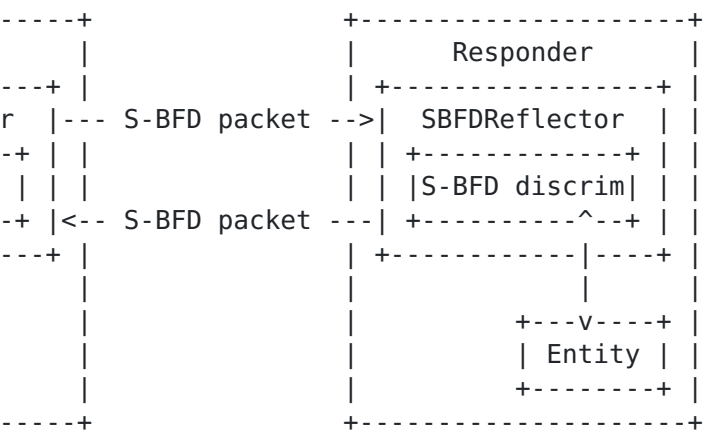


Figure 1: S-BFD Terminology Relationship

3. Seamless BFD Overview

An S-BFD module on each network node allocates one or more S-BFD discriminators for local entities, and creates a reflector BFD session. Allocated S-BFD discriminators may be advertised by applications (ex: OSPF/IS-IS). Required result is that applications, on other network nodes, possess the knowledge of the mapping from remote entities to S-BFD discriminators. The reflector BFD session is to, upon receiving an S-BFD packet targeted to one of local S-BFD discriminator values, transmit a response S-BFD packet back to the initiator.

Once above setup is complete, any network nodes, having the knowledge of the mapping from a remote entity to an S-BFD discriminator, can quickly perform a connectivity test to the remote entity by simply sending S-BFD packets with corresponding S-BFD discriminator value in the "your discriminator" field.

For example:

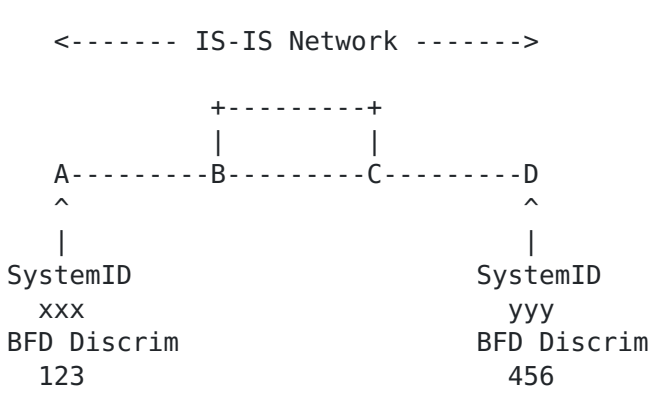


Figure 2: S-BFD for IS-IS Network

The IS-IS with SystemID xxx (node A) allocates an S-BFD discriminator 123, and advertises the S-BFD discriminator 123 in an IS-IS TLV. The IS-IS with SystemID yyy (node D) allocates an S-BFD discriminator 456, and advertises the S-BFD discriminator 456 in an IS-IS TLV. A reflector BFD session is created on both network nodes (node A and node D). When network node A wants to check the connectivity to network node D, node A can send an S-BFD packet, destined to node D, with "your discriminator" field set to 456. When the reflector BFD session on node D receives this S-BFD packet, then response S-BFD packet is sent back to node A, which allows node A to complete the connectivity test.

4. S-BFD UDP Port

S-BFD functions on a well-known UDP port: TBD1.

5. S-BFD Discriminators

Locally allocated S-BFD discriminator values for entities may be arbitrary allocated or derived from values provided by applications. These values may be protocol IDs (ex: System-ID, Router-ID) or network targets (ex: IP address). To minimize the collision of discriminator values between BFD and S-BFD, it is RECOMMENDED that discriminator pool be separate for BFD and S-BFD. Even when employing the separate discriminator pool approach, collision is still possible between one S-BFD application to another S-BFD application, that may be using different values and algorithms to derive S-BFD discriminator values. If the two applications are using S-BFD for a same purpose (ex: network reachability), then the colliding S-BFD discriminator value can be shared. If the two applications are using S-BFD for a different purpose, then the

collision must be addressed. How such collisions are addressed is outside the scope of this document.

One important characteristics of an S-BFD discriminator is that it MUST be unique within an administrative domain. If multiple network nodes allocated a same S-BFD discriminator value, then S-BFD packets falsely terminating on a wrong network node can result in a reflector BFD session to generate a response back, due to "your discriminator" matching. This is clearly not desirable. If only IP based S-BFD is considered, then it is possible for the reflector BFD session to require demultiplexing of incoming S-BFD packets with combination of destination IP address and "your discriminator". Then S-BFD discriminator only has to be unique within a local node. However, S-BFD is a generic mechanism defined to run on wide range of environments: IP, MPLS, etc. For other transports like MPLS, because of the need to use non-routable IP destination address, it is not possible for reflector BFD session to demultiplex using IP destination address. With PHP, there may not be any incoming label stack to aid in demultiplexing either. Thus, S-BFD imposes a requirement that S-BFD discriminators MUST be unique within an administrative domain.

6. Reflector BFD Session

Each network node creates one or more reflector BFD sessions. This reflector BFD session is a session which transmits S-BFD packets in response to received S-BFD packets with "your discriminator" having S-BFD discriminators allocated for local entities. Specifically, this reflector BFD session is to have following characteristics:

- MUST NOT transmit any S-BFD packets based on local timer expiry.

- MUST transmit an S-BFD packet in response to a received S-BFD packet having a valid S-BFD discriminator in the "your discriminator" field, unless prohibited by local policies (ex: administrative, security, rate-limiter, etc).

- MUST be capable of sending only two states: UP and ADMINDOWN.

One reflector BFD session may be responsible for handling received S-BFD packets targeted to all locally allocated S-BFD discriminators, or few reflector BFD sessions may each be responsible for subset of locally allocated S-BFD discriminators. This policy is a local matter, and is outside the scope of this document.

Note that incoming S-BFD packets may be IPv4, IPv6 or MPLS based. How such S-BFD packets reach an appropriate reflector BFD session is also a local matter, and is outside the scope of this document.

7. State Variables

S-BFD introduces new state variables, and modifies the usage of existing ones.

7.1. New State Variables

A new state variable is added to the base specification in support of S-BFD.

- bfd.SessionType: The type of this session. Allowable values are:
  - SBFDInitiator - an S-BFD session on a network node that performs a connectivity test to a target entity by sending S-BFD packets.
  - SBFDReflector - an S-BFD session on a network node that listens for incoming S-BFD packets to local entities and generates response S-BFD packets.

bfd.SessionType variable MUST be initialized to the appropriate type when an S-BFD session is created.

7.2. State Variable Initialization and Maintenance

Some state variables defined in section 6.8.1 of the BFD base specification need to be initialized or manipulated differently depending on the session type. Ed-Note: Anything else?.

- bfd.DemandMode: This variable MUST be initialized to 1 for session type SBFDInitiator, and MUST be initialized to 0 for session type SBFDReflector.

8. S-BFD Procedures

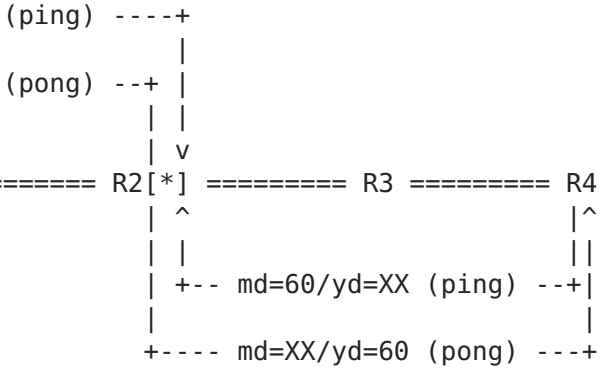
8.1. Initiator Procedures

S-BFD packets transmitted by an SBFDInitiator MUST set "your discriminator" field to an S-BFD discriminator corresponding to the remote entity.

S-BFD packets transmitted by an SBFDInitiator MUST NOT set "my discriminator" field to an S-BFD discriminator allocated for a local entity (and is being monitored by a local SBFDReflector). This is to prevent incoming response S-BFD packets, from a remote SBFDReflector, having "your discriminator" as a S-BFD discriminator of a local entity. Every SBFDInitiator is to have a unique "my discriminator", and SHOULD be allocated from the BFD discriminator pool if the

implementation employs the approach of having separate discriminator pools for BFD and S-BFD.

Below ASCII art describes high level concept of connectivity test using S-BFD. R2 allocates XX as the S-BFD discriminator for its network reachability purpose, and advertises XX to neighbors. ASCII art shows R1 and R4 performing a connectivity test to R2.



[*] Reflector BFD session on R2.
=== Links connecting network nodes.
--- S-BFD packet traversal.

Figure 3: S-BFD Connectivity Test

8.1.1. SBFDInitiator State Machine

An SBFDInitiator may be a persistent session on the initiator with a timer for S-BFD packet transmissions. An SBFDInitiator may also be a module, a script or a tool on the initiator that transmits one or more S-BFD packets "when needed". For transient SBFDInitiators, the BFD state machine described in [RFC5880] may not be applicable. For persistent SBFDInitiators, the states and the state machine described in [RFC5880] will function but are more than necessary. The following diagram provides an optimized state machine for persistent SBFDInitiators. The notation on each arc represents the state of the SBFDInitiator (as received in the State field in the S-BFD packet) or indicates the expiration of the Detection Timer.

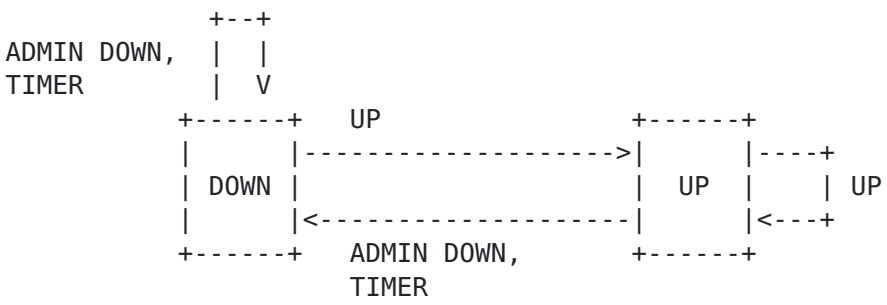


Figure 4: SBFDInitiator FSM

Note that the above state machine is different from the base BFD specification[RFC5880]. This is because the Init state is no longer applicable for the SBFDInitiator. Another important difference is the transition of the state machine from the Down state to the Up state when a packet with State Up is received by the initiator. The definitions of the states and the events have the same meaning as in the base BFD specification [RFC5880].

8.1.2. Details of S-BFD Packet Sent by SBFDInitiator

S-BFD packets sent by an SBFDInitiator is to have following contents:

- Well-known UDP destination port assigned for S-BFD.
- UDP source port as per described in [RFC5881], [RFC5883], [RFC5884] and [RFC5885].
- "my discriminator" assigned by local node.
- "your discriminator" corresponding to a remote entity.
- "State" MUST be set to a value describing local state.
- "Desired Min TX Interval" MUST be set to a value describing local desired minimum transmit interval.
- "Required Min RX Interval" MUST be zero.
- "Required Min Echo RX Interval" SHOULD be zero.
- "Detection Multiplier" MUST be set to a value describing locally used multiplier value.
- Demand (D) bit MUST be set.

8.2. Responder Procedures

A network node which receives S-BFD packets transmitted by an initiator is referred as responder. The responder, upon reception of S-BFD packets, is to perform necessary relevant validations described in [RFC5880], [RFC5881], [RFC5883], [RFC5884] and [RFC5885].

8.2.1. Responder Demultiplexing

A BFD control packet received by a resonder is considered an S-BFD packet if the packet is on the well-known S-BFD port. When a responder receives an S-BFD packet, if the value in the "your discriminator" field is not one of S-BFD discriminators allocated for local entities, then this packet MUST NOT be considered for this mechanism. If the value in the "your discriminator" field is one of S-BFD discriminators allocated for local entities, then the packet is determined to be handled by a reflector BFD session responsible for the S-BFD discriminator. If the packet was determined to be processed further for this mechanism, then chosen reflector BFD session is to transmit a response BFD control packet using procedures described in Section 8.2.2, unless prohibited by local policies (ex: administrative, security, rate-limiter, etc).

8.2.2. Details of S-BFD Packet Sent by SBFDReflector

S-BFD packets sent by an SBFDReflector is to have following contents:

- Well-known UDP destination port assigned for S-BFD.
- UDP source port as described in [RFC5881], [RFC5883], [RFC5884] and [RFC5885].
- "my discriminator" MUST be copied from received "your discriminator".
- "your discriminator" MUST be copied from received "my discriminator".
- "State" MUST be UP or ADMINDOWN. Clarification of reflector BFD session state is described in Section 8.7.
- "Desired Min TX Interval" MUST be copied from received "Desired Min TX Interval".
- "Required Min RX Interval" MUST be set to a value describing how many incoming control packets this reflector BFD session can handle. Further details are described in Section 8.7.
- "Required Min Echo RX Interval" SHOULD be set to zero.
- "Detection Multiplier" MUST be copied from received "Detection Multiplier".
- Demand (D) bit MUST be cleared.

8.3. Diagnostic Values

Diagnostic value in both directions MAY be set to a certain value, to attempt to communicate further information to both ends. However, details of such are outside the scope of this specification.

8.4. The Poll Sequence

Poll sequence MAY be used in both directions. The Poll sequence MUST operate in accordance with [RFC5880]. An SBFDReflector MAY use the Poll sequence to slow down that rate at which S-BFD packets are generated from an SBFDInitiator. This is done by the SBFDReflector using procedures described in Section 8.7 and setting the Poll (P) bit in the reflected S-BFD packet. The SBFDInitiator is to then send the next S-BFD packet with the Final (F) bit set. If an SBFDReflector receives an S-BFD packet with Poll (P) bit set, then the SBFDReflector MUST respond with an S-BFD packet with Poll (P) bit cleared and Final (F) bit set.

8.5. Control Plane Independent (C)

Control plane independent (C) bit for an SBFDInitiator sending S-BFD packets to a reflector BFD session MUST work according to [RFC5880]. Reflector BFD session also MUST work according to [RFC5880]. Specifically, if reflector BFD session implementation does not share fate with control plane, then response S-BFD packets transmitted MUST have control plane independent (C) bit set. If reflector BFD session implementation shares fate with control plane, then response S-BFD packets transmitted MUST NOT have control plane independent (C) bit set.

8.6. Additional SBFDInitiator Behaviors

- If the SBFDInitiator receives a valid S-BFD packet in response to transmitted S-BFD packet to a remote entity, then the SBFDInitiator SHOULD conclude that S-BFD packet reached the intended remote entity.

- When a sufficient number of S-BFD packets have not arrived as they should, the SBFDInitiator SHOULD declare loss of connectivity to the remote entity. The criteria for declaring loss of connectivity and the action that would be triggered as a result are outside the scope of this document.

- Relating to above bullet item, it is critical for an implementation to understand the latency to/from the reflector BFD session on the responder. In other words, for very first S-BFD packet transmitted by the SBFDInitiator, an implementation MUST NOT expect response S-BFD packet to be received for time equivalent to sum of latencies: initiator to responder and responder back to initiator.

- If the SBFDInitiator receives an S-BFD packet with Demand (D) bit set, the packet MUST be discarded.

8.7. Additional SBFDReflector Behaviors

- S-BFD packets transmitted by the SBFDReflector MUST have "Required Min RX Interval" set to a value which expresses how many incoming S-BFD packets this SBFDReflector can handle. The SBFDReflector can control how fast SBFInitiators will be sending S-BFD packets to self by ensuring "Required Min RX Interval" indicates a value based on the current load.

- If the SBFDReflector wishes to communicate to some or all SBFDInitiators that monitored local entity is "temporarily out of service", then S-BFD packets with "state" set to ADMINDOWN are sent to those SBFDInitiators. The SBFDInitiators, upon reception of such packets, MUST NOT conclude loss of connectivity to corresponding remote entity, and MUST back off packet transmission interval for the remote entity to an interval no faster than 1 second. If the SBFDReflector is generating a response S-BFD packet for a local entity that is in service, then "state" in response BFD control packets MUST be set to UP.

- If an SBFDReflector receives an S-BFD packet with Demand (D) bit cleared, the packet MUST be discarded.

9. Scaling Aspect

This mechanism brings forth one noticeable difference in terms of scaling aspect: number of SBFDReflector. This specification eliminates the need for egress nodes to have fully active BFD sessions when only one side desires to perform connectivity tests. With introduction of reflector BFD concept, egress no longer is required to create any active BFD session per path/LSP/function basis. Due to this, total number of BFD sessions in a network is reduced.

10. Co-existence with Traditional BFD

This mechanism has no issues being deployed with traditional BFDs ([RFC5881], [RFC5883], [RFC5884] and [RFC5885]) because S-BFD discriminators which allow this mechanism to function are explicitly reserved and separate UDP port values are used with S-BFD.

11. BFD Echo

BFD echo is outside the scope of this document.

12. Security Considerations

Same security considerations as [RFC5880], [RFC5881], [RFC5883], [RFC5884] and [RFC5885] apply to this document.

Additionally, implementing the following measures will strengthen security aspects of the mechanism described by this document.

- Implementations MUST provide filtering capability based on source IP addresses of received S-BFD packets: [RFC2827].

- Implementations MUST NOT act on received S-BFD packets containing Martian addresses as source IP addresses.

- Implementations MUST ensure that response S-BFD packets generated to the initiator by the SBFDReflector have a reachable target (ex: destination IP address).

- SBFDInitiator MAY pick crypto sequence number based on authentication mode configured.

- SBFDReflector MUST NOT look at the crypto sequence number before accepting the packet.

- SBFDReflector MAY look at the Key ID [I-D.ietf-bfd-generic-crypto-auth] in the incoming packet and verify the authentication data.

- SBFDReflector MUST accept the packet if authentication is successful.

- SBFDReflector MUST compute the Authentication data and MUST use the same sequence number that it received in the S-BFD packet that it is responding to.

- SBFDInitiator MUST accept the S-BFD packet if it either comes with the same sequence number as it had sent or it's within the window that it finds acceptable (described in detail in [I-D.ietf-bfd-generic-crypto-auth])

Using the above method,

- SBFDReflector continue to remain stateless despite using security.

- SBFDReflector are not susceptible to replay attacks as they always respond to S-BFD packets irrespective of the sequence number carried.

- An attacker cannot impersonate the responder since the SBFDInitiator will only accept S-BFD packets that come with the sequence number that it had originally used when sending the S-BFD packet.

13. IANA Considerations

A new value TBD1 is requested from the "Service Name and Transport Protocol Port Number Registry". The requested registry entry is:

```
Service Name (REQUIRED)
  s-bfd
Transport Protocol(s) (REQUIRED)
  udp
Assignee (REQUIRED)
  IESG <iesg@ietf.org>
Contact (REQUIRED)
  BFD Chairs <bfd-chairs@tools.ietf.org>
Description (REQUIRED)
  Seamless Bidirectional Forwarding Detection (S-BFD)
Reference (REQUIRED)
  draft-ietf-bfd-seamless-base
Port Number (OPTIONAL)
  TBD1 (Requesting 7784)
```

14. Acknowledgements

Authors would like to thank Jeffrey Haas for performing thorough reviews and providing number of suggestions. Authors would like to thank Girija Raghavendra Rao, Marc Binderberger, Les Ginsberg, Srihari Raghavan, Vanitha Neelamegam and Vengada Prasad Govindan from Cisco Systems for providing valuable comments. Authors would also like to thank John E. Drake for providing comments and suggestions.

15. Contributing Authors

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Siva Sivabalan
Cisco Systems
Email: msiva@cisco.com

Nagendra Kumar
Cisco Systems
Email: naikumar@cisco.com

Mallik Mudigonda
Cisco Systems
Email: mmudigon@cisco.com

Sam Aldrin
Huawei Technologies
Email: aldrin.ietf@gmail.com

16. References

16.1. Normative References

[I-D.ietf-bfd-seamless-use-case]
Aldrin, S., Bhatia, M., Mirsky, G., Kumar, N., and S. Matsushima, "Seamless Bidirectional Forwarding Detection (BFD) Use Case", draft-ietf-bfd-seamless-use-case-00 (work in progress), June 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, June 2010.

[RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, June 2010.

[RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, June 2010.

16.2. Informative References

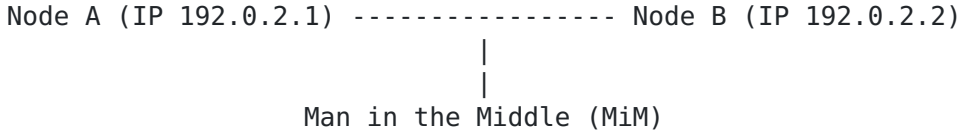
[I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., Zhang, D., and M. Jethanandani, "BFD Generic Cryptographic Authentication", draft-ietf-bfd-generic-crypto-auth-06 (work in progress), April 2014.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC5885] Nadeau, T. and C. Pignataro, "Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)", RFC 5885, June 2010.

Appendix A. Loop Problem

Consider a scenario where we have two nodes and both are S-BFD capable.



Assume node A reserved a discriminator 0x01010101 for target identifier 192.0.2.1 and has a reflector session in listening mode. Similarly node B reserved a discriminator 0x02020202 for its target identifier 192.0.2.2 and also has a reflector session in listening mode.

Suppose MiM sends a spoofed packet with MyDisc = 0x01010101, YourDisc = 0x02020202, source IP as 192.0.2.1 and dest IP as 192.0.2.2. When this packet reaches Node B, the reflector session on Node B will swap the discriminators and IP addresses of the received packet and reflect it back, since YourDisc of the received packet matched with reserved discriminator of Node B. The reflected packet that reached Node A will have MyDdisc=0x02020202 and YourDisc=0x01010101. Since YourDisc of the received packet matched the reserved discriminator of Node A, Node A will swap the discriminators and reflects the packet back to Node B. Since reflectors MUST set the TTL of the reflected packets to 255, the above scenario will result in an infinite loop with just one malicious packet injected from MiM.

FYI: Packet fields do not carry any direction information, i.e., if this is Ping packet or reply packet.

Solutions

The current proposals to avoid the loop problem are:

- Overload "D" bit (Demand mode bit): Initiator always sets the 'D' bit and reflector clears it. This way we can identify if a received packet was a reflected packet and avoid reflecting it back. However this changes the interpretation of 'D' bit.

- Use of State field in the BFD control packets: Initiator will always send packets with State set to "DOWN" and reflector will send back packets with state field set to "UP. Reflectors will

never reflect any received packets with state as "UP". However the only issue is the use of state field differently i.e. state in the S-BFD control packet from initiator does not reflect the local state which is anyway not significant at reflector.

- Use of local discriminator as My Disc at reflector: Reflector will always fill in My Discriminator with a locally allocated discriminator value (not reserved discriminators) and will not copy it from the received packet.

Authors' Addresses

Nobo Akiya
Cisco Systems

Email: nobo@cisco.com

Carlos Pignataro
Cisco Systems

Email: cpignata@cisco.com

Dave Ward
Cisco Systems

Email: wardd@cisco.com

Manav Bhatia
Ionos Networks

Email: manav@ionosnetworks.com

Santosh
Juniper Networks

Email: santoshpk@juniper.net