IPPM
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2021

T. Mizrahi
Huawei Smart Platforms iLab
F. Brockners
S. Bhandari
R. Sivakolundu
C. Pignataro
Cisco
A. Kfir
B. Gafni
Mellanox Technologies, Inc.
M. Spiegel
Barefoot Networks
J. Lemon
Broadcom
October 26, 2020

In-situ OAM Flags
draft-ietf-ippm-ioam-flags-03

Abstract

In-situ Operations, Administration, and Maintenance (IOAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document presents new flags in the IOAM Trace Option headers. Specifically, the document defines the Loopback and Active flags.

Copyright Notice

Table of Contents

1. Introduction

IOAM [I-D.ietf-ippm-ioam-data] is used for monitoring traffic in the network by incorporating IOAM data fields into in-flight data packets.

IOAM data may be represented in one of four possible IOAM options: Pre-allocated Trace Option, Incremental Trace Option, Proof of Transit (POT) Option, and Edge-to-Edge Option. This document defines two new flags in the Pre-allocated and Incremental Trace options: the Loopback and Active flags.

2. Conventions

2.1. Requirement 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 [RFC2119].

2.2. Terminology

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

OAM: Operations, Administration, and Maintenance

3. New IOAM Trace Option Flags

This document defines two new flags in the Pre-allocated and Incremental Trace options:

Bit 1 "Loopback" (L-bit). Loopback mode is used to send a copy of a packet back towards the source, as further described in Section 4.

Bit 2 "Active" (A-bit). When set, this indicates that this is an active IOAM packet, where "active" is used in the sense defined in [RFC7799], rather than a data packet. The packet may be an IOAM probe packet, or a replicated data packet (the second and third use cases of Section 5).

4. Loopback in IOAM

Loopback is used for triggering each transit device along the path to loop back a copy of the data packet. Loopback allows an IOAM encapsulating node to trace the path to a given destination, and to receive per-hop data about both the forward and the return path. Loopback is intended to provide an accelerated alternative to Traceroute, that allows the encapsulating node to receive responses from multiple transit nodes along the path in less then one round-trip-time, and by sending a single packet.

Loopback can be used only if a return path from transit nodes and destination nodes towards the source (encapsulating node) exists. Specifically, loopback is only applicable in encapsulations in which the identity of the encapsulating node is available in the encapsulation header. If an encapsulating node receives a looped back packet that was not originated from the current encapsulating node, the packet is dropped.

The encapsulating node either generates synthetic packets with an IOAM trace option that has the loopback flag set, or sets the loopack flag in a subset of the in-transit data packets. Loopback is used either proactively or on-demand, i.e., when a failure is detected. The encapsulating node also needs to ensure that sufficient space is available in the IOAM header for loopback operation, which includes transit nodes adding trace data on the original path and then again on the return path.

An IOAM trace option that has the loopback bit set MUST have the value '1' in the most significant bit of IOAM-Trace-Type, and '0' in the rest of the bits of IOAM-Trace-Type. Thus, every transit node that processes this trace option only adds a single data field, which is the Hop_Lim and node_id data field. The reason for allowing a single data field per hop is to minimize the impact of amplification attacks.

A loopback bit that is set indicates to the transit nodes processing this option that they are to create a copy of the received packet and send the copy back to the source of the packet. In this context the source is the IOAM encapsulating node, and it is assumed that the source address is available in the encapsulation header. Thus, the source address of the original packet is used as the destination address in the copied packet. The address of the node performing the copy operation is used as the source address. The IOAM transit node pushes the required data field *after* creating the copy of the packet, in order to allow any egress-dependent information to be set based on the egress of the copy rather than the original packet. The copy is also truncated, i.e., any payload that resides after the IOAM option(s) is removed before transmitting the looped back packet back towards the encapsulating node. The original packet continues towards its destination. The L-bit MUST be cleared in the copy of the packet that a node sends back towards the source.

On its way back towards the source, the copied packet is processed like any other packet with IOAM information, including adding any requested data at each transit node (assuming there is sufficient space).

Once the return packet reaches the IOAM domain boundary, IOAM decapsulation occurs as with any other packet containing IOAM information. Note that the looped back packet does not have the L-bit set. The IOAM encapsulating node that initiated the original loopback packet recognizes a received packet as an IOAM looped-back packet by checking the Node ID in the Hop_Lim/node_id field that corresponds to the first hop. If the Node ID matches the current IOAM node, it indicates that this is a looped back packet that was initiated by the current IOAM node, and processed accordingly. If

there is no match in the Node ID, the packet is processed like a conventional IOAM-encapsulated packet.

Note that an IOAM encapsulating node may either be an endpoint (such as an IPv6 host), or a switch/router that pushes a tunnel encapsulation onto data packets. In both cases, the functionality that was described above avoids IOAM data leaks from the IOAM domain. Specificallly, if an IOAM looped-back packet reaches an IOAM boundary node that is not the IOAM node that initiated the loopback, the node does not process the packet as a loopback; the IOAM encapsulation is removed, and since the packet does not have any payload it is terminated. In either case, when the packet reaches the IOAM boundary its IOAM encapsulation is removed, preventing IOAM information from leaking out from the IOAM domain.

5. Active Measurement with IOAM

Active measurement methods [RFC7799] make use of synthetically generated packets in order to facilitate the measurement. This section presents use cases of active measurement using the IOAM Active flag.

The active flag indicates that a packet is used for active measurement. An IOAM decapsulating node that receives a packet with the Active flag set in one of its Trace options must terminate the packet. The active flag is intended to simplify the implementation of decapsulating nodes by indicating that the packet should not be forwarded further. It is not intended as a replacement for existing active OAM protocols, which may run in higher layers and make use of the active flag.

An example of an IOAM deployment scenario is illustrated in Figure 1. The figure depicts two endpoints, a source and a destination. The data traffic from the source to the destination is forwarded through a set of network devices, including an IOAM encapsulating node, which incorporates one or more IOAM options, a decapsulating node, which removes the IOAM options, optionally one or more transit nodes. The IOAM options are encapsulated in one of the IOAM encapsulation types, e.g., [I-D.ietf-sfc-ioam-nsh], or [I-D.ietf-ippm-ioam-ipv6-options].

```
 +--------+     +--------+     +--------+     +--------+     +--------+
 |        |     |  IOAM  |.....|  IOAM  |.....|  IOAM  |     |        |
 +--------+     +--------+     +--------+     +--------+     +--------+
 | L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |<===>| L2/L3  |
 +--------+     +--------+     +--------+     +--------+     +--------+
   Source     Encapsulating    Transit     Decapsulating  Destination
                  Node           Node           Node

                <------------  IOAM domain  ----------->
```

Figure 1: Network using IOAM.

This draft focuses on three possible use cases of active measurement using IOAM. These use cases are described using the example of Figure 1.

- o Endpoint active measurement: synthetic probe packets are sent between the source and destination, traversing the IOAM domain. Since the probe packets are sent between the endpoints, these packets are treated as data packets by the IOAM domain, and do not require special treatment at the IOAM layer. Specifically, the active flag is not used in this case, and the IOAM layer needs not be aware that an active measurement mechanism is used at a higher layer.

- o IOAM active measurement using probe packets within the IOAM domain: probe packets are generated and transmitted by the IOAM encapsulating node, and are expected to be terminated by the decapsulating node. IOAM data related to probe packets may be exported by one or more nodes along its path, by an exporting protocol that is outside the scope of this document (e.g., [I-D.spiegel-ippm-ioam-rawexport]). Probe packets include a Trace Option which has its Active flag set, indicating that the decapsulating node must terminate them.

- o IOAM active measurement using replicated data packets: probe packets are created by the encapsulating node by selecting some or all of the en route data packets and replicating them. A selected data packet that is replicated, and its (possibly truncated) copy is forwarded with one or more IOAM option, while the original packet is forwarded normally, without IOAM options. To the extent possible, the original data packet and its replica are forwarded through the same path. The replica includes a Trace Option that has its Active flag set, indicating that the decapsulating node should terminate it. It should be noted that the current document defines the role of the active flag in allowing the decapsulating

node to terminate the packet, but the replication functionality in this context is outside the scope of this document.

6. IANA Considerations

IANA is requested to allocate the following bits in the "IOAM Trace Flags Registry" as follows:

Bit 1 "Loopback" (L-bit)

Bit 2 "Active" (A-bit)

Note that bit 0 is the most significant bit in the Flags Registry.

7. Performance Considerations

Each of the flags that are defined in this document may have performance implications. When using the loopback mechanism a copy of the data packet is sent back to the sender, thus generating more traffic than originally sent by the endpoints. Using active measurement with the active flag requires the use of synthetic (overhead) traffic.

Each of the mechanisms that use the flags above has a cost in terms of the network bandwidth, and may potentially load the node that analyzes the data. Therefore, it MUST be possible to use each of the mechanisms on a subset of the data traffic; an encapsulating node needs to be able to set the Loopback and Active flag selectively, in a way that considers the effect on the network performance. Similarly, transit and decapsulating nodes need to be able to selectively loop back packets with the Loopback flag, and to selectively export packets. Specifically, rate limiting can be enabled so as to ensure that the mechanisms are used at a rate that does not significantly affect the network bandwidth, and does not overload the receiving entity (or the source node in the case of loopback).

8. Security Considerations

The security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data]. Specifically, an attacker may try to use the functionality that is defined in this document to attack the network.

An attacker may attempt to overload network devices by injecting synthetic packets that include an IOAM Trace Option with one or more of the flags defined in this document. Similarly, an on-path

attacker may maliciously set one or more of the flags of transit packets.

- o Loopback flag: an attacker that sets this flag, either in synthetic packets or transit packet, can potentially cause an amplification, since each device along the path creates a copy of the data packet and sends it back to the source. The attacker can potentially leverage the loopback flag for a Distributed Denial of Service (DDoS) attack, as multiple devices send looped-back copies of a packet to a single source.

- o Active flag: the impact of synthetic packets with the active flag is no worse than synthetic data packets in which the Active flag is not set. By setting the active flag in en route packets an attacker can prevent these packets from reaching their destination, since the packet is terminated by the decapsulating device; however, note that an on-path attacker may achieve the same goal by changing the destination address of a packet. Another potential threat is amplification; if an attacker causes transit switches to replicate more packets than they are intended to replicate, either by setting the Active flag or by sending synthetic packets, then traffic is amplified, causing bandwidth degredation. As mentioned in Section 5, the specification of the replication mechanism is not within the scope of this document. A specification that defines the replication functionality should also address the security aspects of this mechanism.

In order to mitigate the attacks described above, as described in Section 7 it should be possible for IOAM-enabled devices to selectively apply the mechanisms that use the flags defined in this document to a subset of the traffic, and to limit the performance of synthetically generated packets to a configurable rate; specifically, network devices should be able to limit the rate of: (i) looped-back traffic (at transit nodes), (ii) replicated active packets (at encapsulating nodes), (iii) packets that are exported to a collector (from either encapsulating nodes or transit nodes), and (iv) synthetically generated packets (at encapsulating nodes).

Furthermore, as defined in Section 4, transit nodes that process a packet with the Loopback flag only add a single data field, and truncate any payload that follows the IOAM option(s), thus significanly limiting the possible impact of an amplification attack.

IOAM is assumed to be deployed in a restricted administrative domain, thus limiting the scope of the threats above and their affect. This is a fundamental assumtion with respect to the security aspects of IOAM, as further discussed in [I-D.ietf-ippm-ioam-data].

9. References

9.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-10 (work in progress), July 2020.

[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>.

9.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options]
Bhandari, S., Brockners, F., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., Spiegel, M., Krishnan, S., Asati, R., and M. Smith, "In-situ OAM IPv6 Options", draft-ietf-ippm-ioam-ipv6-options-03 (work in progress), September 2020.

[I-D.ietf-sfc-ioam-nsh]
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-04 (work in progress), June 2020.

[I-D.spiegel-ippm-ioam-rawexport]
Spiegel, M., Brockners, F., Bhandari, S., and R. Sivakolundu, "In-situ OAM raw data export with IPFIX", draft-spiegel-ippm-ioam-rawexport-03 (work in progress), March 2020.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

Authors' Addresses

Tal Mizrahi
Huawei Smart Platforms iLab
Israel

Email: tal.mizrahi.phd@gmail.com

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Ramesh Sivakolundu
Cisco Systems, Inc.
170 West Tasman Dr.
SAN JOSE, CA 95134
U.S.A.

Email: sramesh@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Aviv Kfir
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: avivk@mellanox.com

Barak Gafni
Mellanox Technologies, Inc.
350 Oakmead Parkway, Suite 100
Sunnyvale, CA 94085
U.S.A.

Email: gbarak@mellanox.com

Mickey Spiegel
Barefoot Networks
4750 Patrick Henry Drive
Santa Clara, CA 95054
US

Email: mspiegel@barefootnetworks.com

Jennifer Lemon
Broadcom
270 Innovation Drive
San Jose, CA 95134
US

Email: jennifer.lemon@broadcom.com