Network Working Group
Internet-Draft
Updates: 2460 (if approved)
Intended status: Standards Track
Expires: April 25, 2013

M. Eubanks
AmericaFree.TV LLC
P. Chimento
Johns Hopkins University Applied Physics Laboratory
M. Westerlund
Ericsson

October 22, 2012

UDP Checksums for Tunneled Packets
draft-ietf-6man-udpchecksums-05

Abstract

This document provides an update of the Internet Protocol version 6 (IPv6) specification (RFC2460) to improve the performance of IPv6 in the use case when a tunnel protocol uses UDP with IPv6 to tunnel packets. The performance improvement is obtained by relaxing the IPv6 UDP checksum requirement for suitable tunneling protocol where header information is protected on the "inner" packet being carried. This relaxation removes the overhead associated with the computation of UDP checksums on IPv6 packets used to carry tunnel protocols and thereby improves the efficiency of the traversal of firewalls and other network middleboxes by such protocols. We describe how the IPv6 UDP checksum requirement can be relaxed in the situation where the encapsulated packet itself contains a checksum, the limitations and risks of this approach, and define restrictions on the use of this relaxation to mitigate these risks.

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 April 25, 2013.

Table of Contents

1. Introduction

This work constitutes an update of the Internet Protocol Version 6 (IPv6) Specification [RFC2460], in the use case when a tunnel protocol uses UDP with IPv6 to tunnel packets. With the rapid growth of the Internet, tunneling protocols have become increasingly important to enable the deployment of new protocols. Tunneled protocols can be deployed rapidly, while the time to upgrade and deploy a critical mass of routers, switches and end hosts on the global Internet for a new protocol is now measured in decades. At the same time, the increasing use of firewalls and other security related middleboxes means that truly new tunnel protocols, with new protocol numbers, are also unlikely to be deployable in a reasonable time frame, which has resulted in an increasing interest in and use of UDP-based tunneling protocols. In such protocols, there is an encapsulated "inner" packet, and the "outer" packet carrying the tunneled inner packet is a UDP packet, which can pass through firewalls and other middleboxes filtering that is a fact of life on the current Internet.

Tunnel endpoints may be routers or middleboxes aggregating traffic from a large number of tunnel users, therefore the computation of an additional checksum on the outer UDP packet, may be seen as an unwarranted burden on nodes that implement a tunneling protocol, especially if the inner packet(s) are already protected by a checksum. In IPv4, there is a checksum on the IP packet itself, and the checksum on the outer UDP packet can be set to zero. However in IPv6 there is not a checksum on the IP packet and RFC 2460 [RFC2460] explicitly states that IPv6 receivers MUST discard UDP packets with a zero checksum. So, while sending a UDP packet with a zero checksum is permitted in IPv4 packets, it is explicitly forbidden in IPv6 packets. To improve support for IPv6 UDP tunnels, this document updates RFC 2460 to allow tunnel endpoints to use a zero UDP checksum under constrained situations (IPv6 tunnel transports that carry checksum-protected packets), following the considerations in [I-D.ietf-6man-udpzero].

Unicast UDP Usage Guidelines for Application Designers [RFC5405] should be consulted when reading this specification. It discusses both UDP tunnels (Section 3.1.3) and the usage of Checksums (Section 3.4).

While the origin of this specification is the problem raised by the draft titled "Automatic IP Multicast Without Explicit Tunnels", also known as "AMT," [I-D.ietf-mboned-auto-multicast] we expect it to have wide applicability. Since the first version of this document, the need for an efficient UDP tunneling mechanism has increased. Other IETF Working Groups, notably LISP [I-D.ietf-lisp] and Softwires

[RFC5619] have expressed a need to update the UDP checksum processing in RFC 2460. We therefore expect this update to be applicable in future to other tunneling protocols specified by these and other IETF Working Groups.

2. Some Terminology

For the remainder of this document, we discuss only IPv6, since this problem does not exist for IPv4. Therefore all reference to 'IP' should be understood as a reference to IPv6.

The document uses the terms "tunneling" and "tunneled" as adjectives when describing packets. When we refer to 'tunneling packets' we refer to the outer packet header that provides the tunneling function. When we refer to 'tunneled packets' we refer to the inner packet, i.e., the packet being carried in the tunnel.

2.1. 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. Problem Statement

This document provides an update for the case where a tunnel protocol transports tunneled packets that already have a transport header with a checksum. There is both a benefit and a cost to computing and checking the UDP checksum of the outer (encapsulating) UDP transport header. In certain cases, where reducing the forwarding cost is important, such as for systems that perform the check in software, the cost may outweigh the benefit; this document describes a means to avoid that cost. In the case where there is an inner header with a checksum.

4. Discussion

IPv6 UDP Checksum Considerations [I-D.ietf-6man-udpzero] describes the issues related to allowing UDP over IPv6 to have a valid checksum of zero and is not repeated here.

Section 5 and 6 of [I-D.ietf-6man-udpzero], identifies node and inner protocol requirements respectively that introduce constraints on the usage of a zero checksum for UDP over IPv6. This document is intended to satisfy these requirements.

[I-D.ietf-6man-udpzero] and mailing list discussions have noted there is still the possibility of deep-inspection firewall devices or other middleboxes checking the UDP checksum field of the outer packet and thereby discarding the tunneling packets. This would be an issue also for any legacy IPv6 system that has not implemented this update to the IPv6 specification. In this case, the system (according to RFC 2460) will discard the zero-checksum UDP packets, and should log this as an error.

The points below discuss how path errors can be detected and handled in an UDP tunneling protocol when the checksum protection is disabled. Note that other (non-tunneling) protocols may have different approaches, but these are not the topic of this update. We propose the following approach to handle this problem:

- Context (i.e. tunneling state) should be established via application Protocol Data Units (PDUs) that are carried in checksummed UDP packets. That is, any control packets flowing between the tunnel endpoints should be protected by UDP checksums. The control packets can also contain any negotiation required to enable the endpoint/adapters to accept UDP packets with a zero checksum. The control packets may also carry any negotiation required to enable the endpoint/adapters to identify the set of ports that need to enable reception of UDP datagrams with a zero checksum.

- A system never sets the UDP checksum to zero in packets that do not contain tunneled packets.

- UDP keep-alive packets with checksum zero can be sent to validate paths, given that paths between tunnel endpoints can change and so middleboxes in the path may vary during the life of the association. Paths with middleboxes that are intolerant of a UDP checksum of zero will drop the keep-alives and the endpoints will discover that. Note that this need only be done per tunnel endpoint pair, not per tunnel context. Keep-alive traffic can include both packets with tunnel checksums and packets with checksums equal to zero to enable the remote end to distinguish between path failures and the blockage of packets with checksum equal to zero.

- Corruption of the encapsulating IPv6 source address, destination address and/or the UDP source port, and destination port fields : If the restrictions in [I-D.ietf-6man-udpzero] are followed, the inner packets (tunneled packets) will be protected and run the usual (presumably small) risk of having undetected corruption(s). If tunneling protocol contexts contain (at a minimum) source and destination IP addresses and source and destination ports, there

are 16 possible corruption outcomes. We note that these outcomes are not equally likely. The possible corruption outcomes may be:

* Half of the 16 possible corruption combinations have a corrupted destination address. If the incorrect destination is reached and the node doesn't have an application for the destination port, the packet will be dropped. If the application at the incorrect destination is the same tunneling protocol and if it has a matching context (which can be assumed to be a very low probability event) the inner packet will be decapsulated and forwarded. Application developers can verify the context of the packets they receive using UDP, as described in [RFC5405]. Applications that verify the context of a datagram are expected to have a high probability of discarding corrupted data. [I-D.ietf-6man-udpzero] presents examples of cases where corruption can inadvertently impact application state.

* Half of the 8 possible corruption combinations with a correct destination address have a corrupted source address. If the tunnel contexts contain all elements of the address-port 4-tuple, then the likelihood is that this corruption will be detected. It may in fact be discarded on route due to source address validation techniques, such as Unicast Reverse Path Forwarding [RFC2827].

* Of the remaining 4 possibilities, with valid source and destination IPv6 addresses, one has all 4 fields valid, the other three have one or both ports corrupted. Again, if the tunneling endpoint context contains sufficient information, these errors should be detected with high probability.

o Corruption of source-fragmented encapsulating packets: In this case, a tunneling protocol may reassemble fragments associated with the wrong context at the right tunnel endpoint, or it may reassemble fragments associated with a context at the wrong tunnel endpoint, or corrupted fragments may be reassembled at the right context at the right tunnel endpoint. In each of these cases, the IPv6 length of the encapsulating header may be checked (though [I-D.ietf-6man-udpzero] points out the weakness in this check). In addition, if the encapsulated packet is protected by a transport (or other) checksum, these errors can be detected (with some probability).

While they do not guarantee correctness, these mechanism can reduce the risks of relaxing the UDP checksum requirement for IPv6.

5. The Zero-Checksum Update

This specification updates IPv6 to allow a UDP checksum of zero for the outer encapsulating packet of a tunneling protocol. UDP endpoints that implement this update MUST change their behavior for any destination port explicitly configured for zero checksum and MUST NOT discard UDP packets received with a checksum value of zero on the outer packet. When this is done, it requires the constraints in Section 5 and 6 of [I-D.ietf-6man-udpzero].

Specifically, the text in [RFC2460] Section 8.1, 4th bullet is updated. We refer to the following text:

"Unlike IPv4, when UDP packets are originated by an IPv6 node, the UDP checksum is not optional. That is, whenever originating a UDP packet, an IPv6 node must compute a UDP checksum over the packet and the pseudo-header, and, if that computation yields a result of zero, it must be changed to hex FFFF for placement in the UDP header. IPv6 receivers must discard UDP packets containing a zero checksum, and should log the error."

This item should be taken out of the bullet list and should be replaced by:

> Whenever originating a UDP packet, an IPv6 node SHOULD compute a UDP checksum over the packet and the pseudo-header, and, if that computation yields a result of zero, it must be changed to hex FFFF for placement in the UDP header. IPv6 receivers SHOULD discard UDP packets containing a zero checksum, and SHOULD log the error. However, some protocols, such as tunneling protocols that use UDP as a tunnel encapsulation, MAY omit computing the UDP checksum of the encapsulating UDP header and set it to zero, subject to the constraints described in Applicability Statement for the use of IPv6 UDP Datagrams with Zero Checksums [I-D.ietf-6man-udpzero]. In cases where the encapsulating protocol uses a zero checksum for UDP, the receiver of packets sent to a port enabled to receive zero-checksum packets MUST NOT discard packets solely for having a UDP checksum of zero. Note that these constraints apply only to encapsulating protocols that omit calculating the UDP checksum and set it to zero. An encapsulating protocol can always choose to compute the UDP checksum, in which case, its behavior is not updated and uses the method specified in Section 8.1 of RFC2460.
>
> Middleboxes MUST allow IPv6 packets with UDP checksum equal to zero to pass. Implementations of middleboxes MAY allow configuration of specific port ranges for which a zero UDP checksum is valid and may drop IPv6 UDP packets outside those

ranges.

The path between tunnel endpoints can change, thus also the middleboxes in the path may vary during the life of the association. Paths with middleboxes that are intolerant of a UDP checksum of zero will drop any keep-alives sent to validate the path using checksum zero and the endpoints will discover that. Therefore keep-alive traffic SHOULD include both packets with tunnel checksums and packets with checksums equal to zero to enable the remote end to distinguish between path failures and the blockage of packets with checksum equal to zero. Note that path validation need only be done per tunnel endpoint pair, not per tunnel context.

6. Additional Observations

The existence of this issue among a significant number of protocols being developed in the IETF motivates this specified change. The authors would also like to make the following observations:

- An empirically-based analysis of the probabilities of packet corruptions (with or without checksums) has not (to our knowledge) been conducted since about 2000. It is now 2012. We strongly suggest that an empirical study is in order, along with an extensive analysis of IPv6 header corruption probabilities.

- A key cause to the increased usage of UDP in tunneling is the lack of protocol support in middleboxes. Specifically, new protocols, such as LISP [I-D.ietf-lisp], prefer to use UDP tunnels to traverse an end-to-end path successfully and avoid having their packets dropped by middleboxes. If this were not the case, the use of UDP-lite [RFC3828] might become more viable for some (but not necessarily all) tunneling protocols.

- Another issue is that the UDP checksum is overloaded with the task of protecting the IPv6 header for UDP flows (as is the TCP checksum for TCP flows). Protocols that do not use a pseudo-header approach to computing a checksum or CRC have essentially no protection from mis-delivered packets.

7. IANA Considerations

This document makes no request of IANA.

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

8. Security Considerations

It requires less work to generate zero-checksum attack packets than ones with full UDP checksums. However, this does not lead to any significant new vulnerabilities as checksums are not a security measure and can be easily generated by any attacker. Properly configured tunnels should check the validity of the inner packet and perform any needed security checks, regardless of the checksum status. Most attacks are generated from compromised hosts which automatically create checksummed packets (in other words, it would generally be more, not less, effort for most attackers to generate zero UDP checksums on the host).

9. Acknowledgements

We would like to thank Brian Haberman and Gorry Fairhurst for discussions and reviews.

10. References

10.1. Normative References

[I-D.ietf-6man-udpzero]
Fairhurst, G. and M. Westerlund, "Applicability Statement for the use of IPv6 UDP Datagrams with Zero Checksums", draft-ietf-6man-udpzero-07 (work in progress), October 2012.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, July 2004.

[RFC5619] Yamamoto, S., Williams, C., Yokota, H., and F. Parent, "Softwire Security Analysis and Requirements", RFC 5619, August 2009.

10.2. Informative References

[I-D.ietf-lisp]
Farinacci, D., Fuller, V., Meyer, D., and D. Lewis,

"Locator/ID Separation Protocol (LISP)",
draft-ietf-lisp-23 (work in progress), May 2012.

[I-D.ietf-mboned-auto-multicast]
Bumgardner, G., "Automatic Multicast Tunneling",
draft-ietf-mboned-auto-multicast-14 (work in progress),
June 2012.

[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.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines
for Application Designers", BCP 145, RFC 5405,
November 2008.

Authors' Addresses

Marshall Eubanks
AmericaFree.TV LLC
P.O. Box 141
Clifton, Virginia 20124
USA

Phone: +1-703-501-4376
Fax:
Email: marshall.eubanks@gmail.com

P.F. Chimento
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
USA

Phone: +1-443-778-1743
Email: Philip.Chimento@jhuapl.edu

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com