Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 27, 2015

B. Williams
Akamai, Inc.
M. Boucadair
France Telecom
D. Wing
Cisco Systems, Inc.
October 24, 2014

Experimental Option for TCP Host Identification
draft-williams-exp-tcp-host-id-opt-04

Abstract

Recent IETF proposals have identified benefits to more distinctly identifying the hosts that are hidden behind a shared address/prefix sharing device or application-layer proxy. Analysis indicates that the use of a TCP option for this purpose can be successfully applied to a broad range of use cases. This document describes a common experimental TCP option format for host identification.

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 27, 2015.

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

A broad range of issues associated with address sharing have been well documented in [RFC6269] and [I-D.boucadair-intarea-host-identifier-scenarios]. In addition, [RFC6967] provides analysis of various solutions to the problem of revealing the sending host's identifier (HOST_ID) information to the receiver, indicating that a solution using a TCP [RFC0793] option for this purpose is among the possible approaches that could be applied with limited performance impact and a high success ratio. The purpose of this document is to define such a TCP option in order to factilitate further validation of the mechanism.

Multiple recent Internet Drafts define TCP options for the purpose of host identification: [I-D.wing-nat-reveal-option], [I-D.abdo-hostid-tcpopt-implementation], and [I-D.williams-overlaypath-ip-tcp-rfc]. Specification of multiple option formats to serve the purpose of host identification increases the burden for potential implementers and presents interoperability challenges as well. This document defines a common TCP option format that supersedes all three of the above proposals.

The option defined in this document uses the TCP experimental option codepoint sharing mechanism defined in [RFC6994] and is intended to allow broad deployment of the mechanism on the public Internet in order to validate the utility of this option format for the intended use cases.

Section 5 of this document discusses compatibility between this new TCP option and existing commonly deployed TCP options.

1.1. Important Use Cases

This memo focuses primarily on the carrier grade NAT (CGN), application proxy, and overlay network use cases described in [I-D.boucadair-intarea-host-identifier-scenarios]. This means that the option could either be applied to an individual TCP packet at the connection endpoint (e.g. an application proxy or a transport layer overlay network) or at an address-sharing middle box (e.g. a CGN or a network layer overlay network). See Section 4 below for additional details about the types of devices that could add the option to a TCP packet, as well as limitations on use of the option when it is to be

inserted by an address-sharing middlebox, including issues related to packet fragmentation.

The receiver-side use cases considered by this memo include the following:

- Differentiating between attack and non-attack traffic when the source of the attack is sharing an address with non-attack traffic.

- Application of per-client policies for resource utilization, etc. when multiple clients are sharing a common address.

- Improving server-side load-balancing decisions by allowing the load for multiple clients behind a shared address to be assigned to different servers, even when session-affinity is required at the application layer.

In all of the above cases, differentiation between address-sharing clients commonly needs to be performed by a network function that does not process the application layer protocol (e.g. HTTP) or the sercurity protocol (e.g. TLS), because the action needs to be performed prior to decryption or parsing the application layer. Due to this, a solution implemented within the application layer or security protocol cannot fully meet the receiver-side requirements. At the same time, as noted in [RFC6967], use of an IP option for this purpose has a low success rate. For these reasons, using a TCP option to deliver the host identifier has been selected as the most effective way to satisfy these specific use cases.

1.2. Experiment Goals

The extensive testing effort documented in [I-D.abdo-hostid-tcpopt-implementation] confirmed that a TCP option could be used for host identification purposes without significant disruption of TCP connectivity to legacy servers that do no support the option. It also showed how mechanisms available in existing TCP implementations could make use of such a TCP option for improved diagnostics and/or packet filtering.

Specification of the TCP option described in this memo will allow further experiments to be conducted in order to assess the viability of the option for the receiver-side use cases discussed above:

- Differentiate between attack and non-attack traffic.

- Enforce per-client policies.

o Assist load-balancing decision-making.

In particular, real-world deployment of the option is expected to provide opportunities for engagement with a broader range of both application and middleware implementations in order to develop a more complete picture of how well the option meets the use-case requirements.

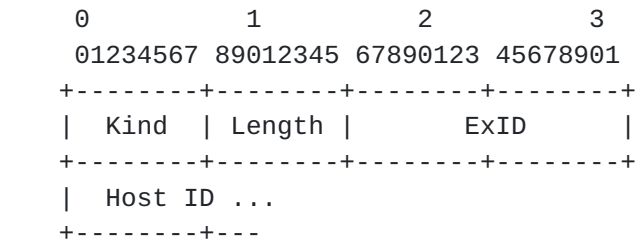
In addition, continued experimentation on the open internet following publication of this memo is expected to allow further refinement of requirements related to the values used to populate the option and how those values can be interpretted by the receiver. There is a tradeoff between providing the expected functionality to the receiver and protecting the privacy of the sender, and additional work is necessary in order to find the right balance.

2. Terminology

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].

3. Option Format

When used for host identification, the TCP experimental option uses the experiment identification mechanism described in [RFC6994] and has the following format and content.



Kind: The option kind value is 253

Length: The length of the option is variable, based on the required size of the host identifier (e.g. a 2 octet host ID will require a length of 6, while a 4 octet host ID will require a length of 8).

ExID: The experiment ID value is 0x0348 (840).

Host ID: The host identifier is an application dependent value with an interpretation agreed upon by the sender and the receiver.

When multiple host identifiers are necessary (e.g. a list of IP addresses, an IP address and a port number), the HOST_ID option is included multiple times within the packet, once for each identifier. While this approach significantly increases option space utilization when multiple identifiers are included, cases where only a single identifier is included are more common and thus it is beneficial to optimize for those cases.

4. Option Use

This section describes requirements associated with the use of the option, including: which hosts are allowed to include the option, expected option values, and segments that include the option.

4.1. Sending Host Requirements

The HOST_ID option MUST only be added by the sending host or any device involved in the forwarding path that changes IP addresses and/or TCP port numbers (e.g., NAT44 [RFC3022], Layer-2 Aware NAT, DS-Lite AFTR [RFC6333], NPTv6 [RFC6296], NAT64 [RFC6146], Dual-Stack Extra Lite [RFC6619], TCP Proxy, etc.). The HOST_ID option MUST NOT be added or modified en-route by any device that does not modify IP addresses and/or TCP port numbers.

4.2. Option Value Requirements

The information conveyed in the HOST_ID option is intended to uniquely identify the sending host to the best capability of the machine that adds the option to the segment, while at the same time avoiding inclusion of information that does not assist this purpose. In addition, the option is not intended to be used to expose information about the sending host that could not be discovered by observing segments in transit on some portion of the internet path between the sender and the receiver. As noted in Section 1.2, identifying the optimal set of values to use for this purpose is one of the experiment goals for this document. For this reason, the document attempts to provide a high degree of flexibility for the machine that adds the option to TCP segments.

The HOST_ID option value MUST correlate to IP addresses and/or TCP port numbers that were changed by the inserting host/device (i.e., some of the IP address and /or port number bits are used to generate the HOST_ID).

Intermediary devices (e.g. address sharing device) SHOULD be configurable to enable including the HOST_ID TCP option. These devices MUST be configured with the type of information to populate the HOST_ID TCP option (e.g. certain bits of the source IPv6 address, the full source IPv6 address, certain bits of the source IPv4 address, the full source IPv4 address, the source port number, etc.).

The device MAY be configured to include multiple identifiers (e.g. both a source IP address and a source port number). In such case, the device MUST insert two instances of the HOST_ID option, each of which contains the appropriate information. Note, there is no need to signal the semantic of the included data as this specification assumes the service is aware of that information by out of band means (e.g. both the service and the address sharing device are managed by the same administrative entity).

The device MUST be configured with the behavior to follow when a HOST_ID TCP option is already present in the segment:

- If the device is configured to strip any existing HOST_ID TCP option, it MUST remove all occurrences of the HOST_ID in a received TCP segment.

- If the device is configured to strip existing HOST_ID TCP options and insert a local HOST_ID TCP Option, it MUST remove all occurrences of the HOST_ID in a received TCP segment and then MUST include a local HOST_ID TCP option. The device MAY be configured to use existing HOST_ID TCP options as differentiators when selecting the value to use in the local HOST_ID TCP option.

- The device MAY be configured to maintain any existing HOST_ID TCP option(s) in the received segments, the device MUST NOT remove those instances of the option. Furthermore, it MUST add a new HOST_ID TCP option while preserving the order of appearance in the TCP option space. In particular, the local HOST_ID TCP option MUST appear as the last occurrence of the HOST_ID TCP option in the segment.

  Note: Because the order of appearance of TCP options could be modified by some middleboxes, deployments MUST NOT rely on option order to provide additional meaning to the individual options. Instead, as indicated above, the full set of option values, with their lengths, MUST be treated as a single unified identifier.

4.3. Segment Inclusion Requirements

A sending host or intermediary device that is configured to include the HOST_ID option MUST include the option in SYN segments.

The sending host or intermediary device cannot determine whether the option value is used in a stateful manner by the receiver, nor can it determine whether SYN cookies are in use by the receiver. For this reason, the option MUST be included in all segments until return segments from the receiver positively indicate that the TCP connection is fully established on the receiver (e.g. the return segment either includes or acknowledges data).

4.3.1. Alternative SYN Cookie Support

The authors have also considered an alternative approach to SYN cookie support in which the receiving host (i.e. the host that accepts the TCP connection) to echo the option back to the sender in the SYN/ACK segment when a SYN cookie is being sent. This would allow the sending host to determine whether further inclusion of the option is necessary. This approach would have the benefit of not requiring inclusion of the option in non-SYN packets if SYN cookies had not been used. Unfortunately, this approach fails if the sending host itself does not support the option, since an intermediate node would have no way to determine that SYN cookies had been used.

4.3.2. Packet Fragmentation

The option SHOULD NOT be included in packets if the resulting packet would require local fragmentation.

5. Interaction with Other TCP Options

This section details how the HOST_ID option functions in conjunction with other TCP options.

5.1. Option Space

TCP provides for a maximum of 40 octets for TCP options. As discussed in Appendix A of Multipath TCP (MPTCP) [RFC6824], a typical SYN from modern, popular operating systems contain several TCP options (MSS, window scale, SACK permitted, and timestamp) which consume 19-24 octets depending on word alignment of the options. The initial SYN from a multipath TCP client would consume an additional 16 octets.

HOST_ID needs at least 6 octets to be useful, so 9-21 octets are

sufficient for many scenarios that benefit from HOST_ID. However, 4 octets are not enough space for the HOST_ID option. Thus, a TCP SYN containing all the typical TCP options (MSS, window Scale, SACK permitted, timestamp), and also containing multipath capable or multipath join, and also being word aligned, has insufficient space to also accommodate HOST_ID. This means something has to give. The choices are to avoid word alignment in that case (freeing 5 octets), remove a TCP option from the original TCP SYN, or avoid adding the HOST_ID option. We expect to learn from deployment experience during the experiment which of these options, or a combination of these options, is best.

5.2. Authentication Option (TCP-AO)

The TCP-AO option [RFC5925] supports a "TCP option flag" to indicate whether TCP options other than TCP-AO are included in the MAC calculation (Section 3.1 of [RFC5925]). When the options are not included in the MAC calculation, the use of HOST_ID option does not interfere with TCP-AO option. However, because TCP-AO provides integrity protection of the source IP address, TCP-AO is broken in the presence of NAT.

Because TCP-AO is incompatible with address sharing, an experimental extension to TCP-AO (called TCP-AO-NAT) is introduced in [RFC6978]. Injecting a HOST_ID TCP option does not interfere with the use of TCP-AO-NAT because the TCP options are not included in the MAC calculation.

6. Security Considerations

Security (including privacy) considerations common to all HOST_ID solutions are discussed in [RFC6967].

The content of the HOST_ID option MUST NOT be used for purposes that require a trust relationship between the sender and the receiver (e.g. billing and/or intrusion prevention) unless a mechanism outside the scope of this specification is used to ensure the necessary level of trust.

When the receiving network uses the values provided by the option in a way that does not require trust (e.g. maintaining session affinity in a load-balancing system), then use of a mechanism to enforce the trust relationship is OPTIONAL.

7. Privacy Considerations

Sending a TCP SYN across the public Internet necessarily discloses the public IP address of the sending host. When an intermediate address sharing device is deployed on the public Internet (see [I-D.boucadair-intarea-host-identifier-scenarios] for examples), anonymity of the hosts using the device will be increased, with hosts represented by multiple source IP addresses on the ingress side of the device using a single source IP address on the egress side. The HOST_ID TCP option removes that increased anonymity, taking information that was already visible in TCP packets on the public Internet on the ingress side of the address sharing device and making it available on the egress side of the device as well. In some cases, an explicit purpose of the address sharing device is anonymity, in which case use of the HOST_ID TCP option would be incompatible with the purpose of the device.

The HOST_ID option MUST NOT be used to provide client geographic or network location information that was not publicly visible in IP packets for the TCP flows processed by the inserting host. For example, the client's IP address MAY be used as the HOST_ID option value, but any geographic or network location information derived from the client's IP address MUST NOT be used as the HOST_ID value.

The HOST_ID option MAY provide differentiating information that is locally unique such that individual TCP flows processed by the inserting host can be reliably identified. The HOST_ID option MUST NOT provide client identification information that was not publicly visible in IP packets for the TCP flows processed by the inserting host.

The HOST_ID option MUST be stripped from IP packets traversing middle boxes that provide network-based anonymity services.

8. IANA Considerations

This document specifies a new TCP option that uses the shared experimental options format [RFC6994], with ExID=0x0348 (840) in network-standard byte order. This ExID has already been registered with IANA.

9. Acknowledgements

Many thanks to J. Touch, M. Scharf, W. Eddy, T. Reddy, and Y. Nishida for their comments.

10. References

10.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

10.2. Informative References

[I-D.abdo-hostid-tcpopt-implementation]
Abdo, E., Boucadair, M., and J. Queiroz, "HOST_ID TCP Options: Implementation & Preliminary Test Results", draft-abdo-hostid-tcpopt-implementation-03 (work in progress), July 2012.

[I-D.boucadair-intarea-host-identifier-scenarios]
Boucadair, M., Binet, D., Durel, S., Chatras, B., Reddy, T., Williams, B., Sarikaya, B., Xue, L., and R. Wheeldon, "Scenarios with Host Identification Complications", draft-boucadair-intarea-host-identifier-scenarios-07 (work in progress), July 2014.

[I-D.williams-overlaypath-ip-tcp-rfc]
Williams, B., "Overlay Path Option for IP and TCP", draft-williams-overlaypath-ip-tcp-rfc-04 (work in progress), June 2013.

[I-D.wing-nat-reveal-option]
Yourtchenko, A. and D. Wing, "Revealing hosts sharing an IP address using TCP option", draft-wing-nat-reveal-option-03 (work in progress), December 2011.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P.

Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6619] Arkko, J., Eggert, L., and M. Townsley, "Scalable Operation of Address Translators with Per-Interface Bindings", RFC 6619, June 2012.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

[RFC6967] Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Potential Solutions for Revealing a Host Identifier (HOST_ID) in Shared Address Deployments", RFC 6967, June 2013.

[RFC6978] Touch, J., "A TCP Authentication Option Extension for NAT Traversal", RFC 6978, July 2013.

[RFC6994] Touch, J., "Shared Use of Experimental TCP Options", RFC 6994, August 2013.

Appendix A. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

A.1. Changes from version 03 to 04

Improve discussion of RFC6967.

Don't use "message" to describe TCP segments.

Add reference to RFC6994 to section 3.

Clarify that this draft superseeds earlier drafts.

Improve discussion of SYN cookie handling.

Remove lower case uses of keywords (e.g. must, should, etc.)

throughout the document.

Some stronger privacy guidance, replacing SHOULD with MUST.

Add an experiment goal related to optimal option value.

Add text related to the identification goals of the option value (still needs more work).

A.2. Changes from version 02 to 03

Clarification of arguments in favor of this approach.

Add discussion of important use cases.

Clarification of experiment goals and earlier test results.

A.3. Changes from version 01 to 02

Add note re: order of appearance.

A.4. Changes from version 00 to 01

Add discussion of experiment goals.

Limit external references to the earlier drafts.

Add guidance to limit the types of device that add the option.

Improve/correct discussion of TCP-AO and security.

Appendix B. Open Issues

[Note to RFC Editor: Please remove this section prior to publication.]

Add discussion of non-local fragmentation.

Evaluate the reliability of attempts to exclude the option when local fragmentation would be required.

Clarify exactly what the identifier is identifying.

Improve discussion on interpretation of multiple instances of the option, including order of interpretation and set interpretation.

Evaluate whether use of multiple identifiers should be constrained.

Discuss the possibility of the option value changing over the life of the connection.

Clarify use cases related to stripping and replacing the option.

Make this draft self-contained, rather than referring readers to use-cases and requirements contained in other I.D.s that were never published as RFCs.

Add discussion of TCP Fast Open.

Add experiment goal related to identifying methods for receiver-side use of data conveyed in the option.

Re-evaluate all use of MUST, MAY, SHOULD thoughout the document.

Clarify use of SHOULD rather than MUST where possible, or perhaps generally.

Correct some discussion of TCP-AO and TCP-AO-NAT.

Clarify the security requirements re: trust relationship.

Clarify privacy considerations regarding NATs that separate private and public networks.

Remove restatement of requirements from other documents.

Authors' Addresses

Brandon Williams
Akamai, Inc.
8 Cambridge Center
Cambridge, MA 02142
USA

Email: brandon.williams@akamai.com

Mohamed Boucadair
France Telecom
Rennes, 35000
Fance

Email: mohamed.boucadair@orange.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: dwing@cisco.com