Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 10, 2012

A. Yourtchenko
D. Wing
cisco

December 8, 2011

Revealing hosts sharing an IP address using TCP option draft-wing-nat-reveal-option-03

Abstract

When an IP address is shared among several subscribers -- with a NAT or with an application-level proxy -- it is impossible for the server to differentiate between different clients. Such differentiation is valuable in several scenarios. This memo describes a technique to differentiate TCP clients sharing an IP address. The proposed method uses a TCP option, which avoids altering the application-level payload and works well with SSL-protected connections.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 10, 2012.

Table of Contents

1. Introduction

When clients are allocated unique, publicly-routable IPv4 addresses, it is easy to associate certain characteristics with their IP address. For example, if an IP address sends a lot of spam, that IP address is classified by many public (and private) system as "a spammer". Such classification can cause email or other traffic from that IP address to be blocked, rate limited, challenged with a captcha, or to receive other treatment. Reputation systems of various sorts exist for a wide variety of services on the Internet including IMAP, HTTP, ssh -- often these systems will slow down or interfere with normal login attempts when a dictionary attack is detected. An IP address can be added to a multitude of 'reputation' systems. Some of these systems are distributed across the Internet, some are shared amongst consenting parties, and some are operated by individual enterprises or individual hosts. Further discussion of the impacts of address sharing can be found in [I-D.ietf-intarea-shared-addressing-issues].

With the exhaustion of the IPv4 address space, IPv4 addresses will be shared on a large scale. This sharing will persist long after IPv6 is ubiquitous -- in fact, IPv4 address sharing will persist until all content and services on the Internet are available over IPv6. Once all content and services are available over IPv6, an Internet service provider will no longer need to provide access to the IPv4 Internet.

Until that time, both legitimate users and attackers will share IPv4 addresses. This IP address sharing means legitimate users will share the reputation of attackers.

This document describes a TCP option which can be added by an address sharing device such as a NAT or an application-level proxy. This TCP option allows a TCP server to differentiate between the TCP clients sharing that IP address.

An analysis of other techniques is available in [I-D.boucadair-intarea-nat-reveal-analysis].

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 [RFC2119].

subscriber: the client accessing an address sharing device, who is responsible for the actions of their device(s). This might be an individual handset (with mobile devices), a home Internet connection,

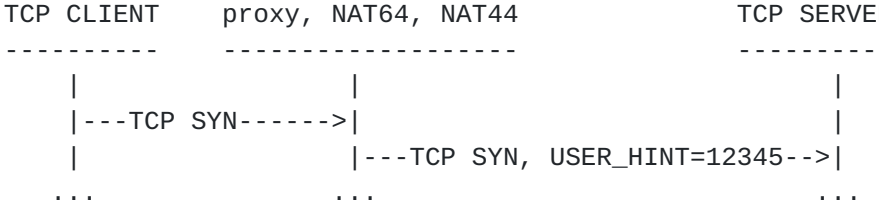
a small-medium business Internet connection, a University dormitory room, an individual employee of a company, or the company itself.

3. Description

This proposal defines one new TCP option, USER_HINT, to contain the TCP client's 16 bit identifier. This value might be the lower 16 bits of their IPv4 address, their VLAN ID, VRF ID, subscriber ID, or similar. The address sharing device (NAT, application proxy) would add the TCP option to the TCP SYN packet. TCP options are treated outside of the TCP sequence space, so no modifications of either sequence or acknowledgement numbers are needed.

3.1. Operation of Address Sharing Device

The address sharing device inserts the USER_HINT option into the TCP SYN, as depicted below. If the TCP SYN already has a USER_HINT option present, it is ignored and over-written with the new value.



3.2. Operation of the TCP Server

The TCP server identifies the client by combining the source IPv4 address in the IP header with the data in the USER_HINT option. This can be implemented by modifying the TCP stack to make the USER_HINT data available to the application via an API (e.g., via a socket option).

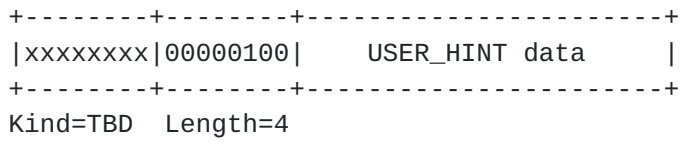
3.3. Reusing the USER_HINT value

The USER_HINT value is only 16 bits, so is obviously not globally unique. Even when combined with the publicly-routable IP address, the additional 16 bits are still not guaranteed to uniquely identify a particular subscriber. Out of necessity, the numbering space will be re-used by some address sharing devices, especially address sharing devices that are sharing many users on one IP address. As with today's IPv4 addresses which are assigned by an ISP, deterministically associating the IPv4 address (or IPv4 address and USER_HINT) with a particular subscriber requires more than simply completing a TCP 3-way handshake. For example, over a single day, an address sharing device might serve tens of thousands of different

subscribers from the same shared IP address, and thus it will need to rotate through the 16 bit USER_HINT space several times during the day. When doing so, the USER_HINT MUST NOT be re-used more often than every 2 minutes (a number chosen out of thin air); if an address sharing device needs to re-use a USER_HINT value more often than that, it should use additional IP addresses (to reduce how quickly the USER_HINT space is consumed on each address) or simply send TCP SYNs without USER_HINT until 2 minutes have elapsed. This 2 minute delay is necessary to allow the reputation system on a TCP server to differentiate between subscribers. For most implementations, the port sharing ratio (rather than a timer) is sufficient to meet this requirement.

4. USER_HINT Option Format

The USER_HINT option is always 4 bytes long, with 16 bits of USER_HINT data



User Hint option data: 16 bits.

If this option is present, it differentiates between active TCP hosts sharing the same IP address. This field MUST only be sent in the initial connection request (i.e., in segments with the SYN control bit set), or in the first ACK if the server's SYN contained the USER_HINT option.

5. Interaction with other TCP Options

This section details how USER_HINT functions in conjunction with other TCP options.

5.1. Option Space

As discussed in Appendix A of Multipath TCP (MPTCP) [I-D.ietf-mptcp-multiaddressed], there is a maximum of 40 bytes for TCP options, and a typical SYN (with MSS, window scale, SACK permitted, and timestamp options) leaves 16 bytes spare (if the options are word-aligned) or 21 bytes spare (if the options are not word-aligned).

Thus, the 4 byte option proposed in this memo would not cause a

problem with a typical TCP SYN.

5.2. Multipath TCP (MPTCP)

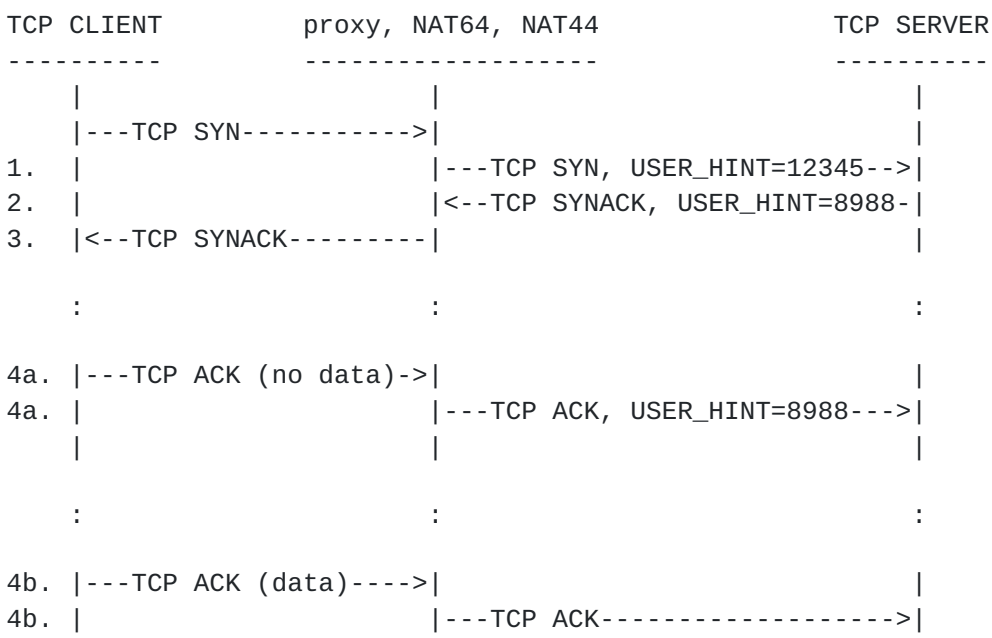
If the TCP client supports Multipath TCP (MPTCP) [I-D.ietf-mptcp-multiaddressed], the client will include the Multipath Capable or Multipath Join options to the TCP SYN. The Multipath Capable (MP_CAPABLE) option consumes 12 bytes, so a SYN containing all of these options would fully consume the 40 byte SYN option space. The Multipath Join (MP_JOIN) can consumes 12 or 16 bytes, but it is only used after successful early exchange containing the MP_CAPABLE option. Thus, there is reason to include USER_HINT if MP_JOIN is present in the TCP SYN -- if the MP_JOIN is not valid, it will be rejected by the server without creating any state on the server. Furthermore, if a client TCP is multi-homed, the client's TCP connections will probably go through different address sharing devices and thus have different externally-visible IP addresses and different USER_HINT values. Thus, it is NOT RECOMMENDED to include the USER_HINT option if the TCP SYN contains the MP_JOIN option.

5.3. Authentication Option (TCP-AO)

The USER_HINT option is incompatible with the Authentication Option (TCP-AO) [RFC5925], because TCP-AO provides integrity protection of the TCP SYN, including TCP options. However, TCP-AO is already incompatible with address sharing, because TCP-AO provides integrity protection of the source IP address.

6. Interaction with TCP SYN Cookies

TCP SYN cookies [RFC4987] are commonly deployed to mitigate TCP SYN attacks, which have some side effects. The USER_HINT information in the TCP SYN provides the TCP server with additional information it can use when deciding if this TCP connection attempt should be answered with a SYN cookie or should be answered normally. In the event the TCP server does not (or cannot) store the USER_HINT data, the USER_HINT data can be re-established on the TCP server when the client's first ACK is sent. There is a slight risk, however, that the client's first ACK, as seen by the middlebox, might contain data. If it does contain data, adding another 4 bytes to the packet could cause MTU to be exceeded.



The procedure is as follows:

1. Upon receiving a TCP SYN containing the USER_HINT option, the TCP server MAY respond to a SYN containing USER_HINT with an ACK packet containing its own USER_HINT value. (Note: this ACK response will typically have the SYN bit set.) If the server does not include the USER_HINT in its ACK packet, processing stops.

2. The middlebox, upon seeing the USER_HINT in the ACK, records those 2 bytes, which are used in a later step.

3. The middlebox strips the USER_HINT from the ACK, so it is not received by the TCP client. The middlebox sends the TCP ACK, without its USER_HINT option, to the TCP client.

4. The TCP client responds normally, generating a TCP ACK.

5. The middlebox receives an ACK from the TCP client. This ACK will either contain:

   A. no data, which causes the middlebox to add the USER_HINT value (from step 2) to the TCP ACK

   B. data, which causes the middlebox to simply forward the ACK packet. This is done to avoid MTU problems between the middlebox and the TCP server.

State is required in the address sharing device to perform the steps described in this section. This isn't a disaster with stateful address sharing (e.g., NAPT). However, in an A+P-like system (e.g., [I-D.ymbk-aplusp], [I-D.despres-intarea-4rd]), the CPE would need to perform the USER_HINT function, which introduces additional security considerations (not yet discussed in this version of the document).

7. Security Considerations

An attacker might use this functionality to appear as if IP address sharing is occurring, in the hopes that a naive server will allow additional attack traffic. TCP servers and applications SHOULD NOT assume the mere presence of the functionality described in this paper indicates there are other (benign) users sharing the same IP address.

8. Acknowledgements

Thanks to Anantha Ramaiah for the discussion. Thanks to Senthil Sivakumar for his review.

9. IANA Considerations

Assign a new TCP option number (kind value), USER_HINT.

10. References

10.1. Normative References

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

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

10.2. Informative References

[I-D.boucadair-intarea-nat-reveal-analysis]
Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Solution Candidates to Reveal a Host Identifier in Shared Address Deployments", draft-boucadair-intarea-nat-reveal-analysis-04 (work in progress), September 2011.

[I-D.despres-intarea-4rd]

Despres, R., Matsushima, S., Murakami, T., and O. Troan, "IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT's made optional", draft-despres-intarea-4rd-01 (work in progress), March 2011.

[I-D.ietf-intarea-shared-addressing-issues] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", draft-ietf-intarea-shared-addressing-issues-05 (work in progress), March 2011.

[I-D.ietf-mptcp-multiaddressed] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", draft-ietf-mptcp-multiaddressed-04 (work in progress), July 2011.

[I-D.ymbk-aplusp] Bush, R., "The A+P Approach to the IPv4 Address Shortage", draft-ymbk-aplusp-10 (work in progress), May 2011.

[RFC4987] Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", RFC 4987, August 2007.

Appendix A. Change History

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

A.1. Changes from draft-wing-nat-reveal-option-01 to -02

- Limit option value to 16 bits (which becomes 32 bits total with the 8 bit option number and 8 bit length)
- described how USER_HINT can work successfully with Multipath TCP (MPTCP)'s options.
- Better described operation with TCP SYN Cookies.
- Renamed option from CX-ID to USER_HINT

Authors' Addresses

Andrew Yourtchenko
Cisco Systems, Inc.
6a de Kleetlaan
Diegem 1831
BE

Phone: +32 2 704 5494
Email: ayourtch@cisco.com

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

Email: dwing@cisco.com