Network Working Group
Internet-Draft
Expires: August 5, 2006

S. Guha
Cornell U.
K. Biswas
Cisco Systems
B. Ford
M.I.T.
P. Francis
Cornell U.
S. Sivakumar
Cisco Systems
P. Srisuresh
Consultant
Feb 2006

NAT Behavioral Requirements for Unicast TCP
draft-hoffman-behave-tcp-04.txt

Status of this Memo

Copyright Notice

Abstract

This document defines a set of requirements for NATs that handle TCP that would allow many applications, such as peer-to-peer applications and on-line games, to work consistently. Developing NATs that meet this set of requirements will greatly increase the likelihood that these applications will function properly.

Table of Contents

1. Applicability Statement 3
2. Introduction 3
3. Terminology 3
4. TCP Session Setup 4
4.1 Address and Port Mapping 4
4.2 Internally Initiated Sessions 4
4.3 Externally Initiated Sessions 5
5. TCP Session Refresh 6
6. Application Level Gateways 7
7. Requirements 7
8. Security considerations 7
9. IANA considerations 9
10. Acknowledgments 9
11. Normative References 9
Authors' Addresses 10
Intellectual Property and Copyright Statements 12

1. Applicability Statement

This document is adjunct to RFC FIXME-BEHAVE-UDP [1], which defines many terms relating to NATs, lays out general requirements for all NATs, and sets requirements for NATs that handle unicast UDP traffic. The purpose of this document is to set requirements for NATs that handle TCP traffic (that is, almost every NAT).

The requirements of this specification apply to Traditional NATs as described in RFC 2663 [2].

This document only covers the TCP aspects of NAT traversal. Firewalls, and packet inspection above the TCP layer are out-of-scope. Middle-box behavior that is not necessary for network address translation of TCP is out-of-scope. Application and OS aspects of TCP NAT traversal are out-of-scope. Signaling based approaches to NAT traversal such as Midcom and UPnP that directly control the NAT are out-of-scope.

2. Introduction

Network Address Translators (NATs) hinder connectivity in applications where connections may be initiated to internal hosts. RFC FIXME-BEHAVE-UDP [1] lays out the terminology and requirements for NATs in the context of UDP. This document supplements these by setting requirements for NATs that handle TCP traffic. All definitions and requirements in [1] are inherited here.

Recently, many techniques have been devised to make peer-to-peer TCP applications work across NATs. STUNT [3], NATBLASTER [4], and P2PNAT [5] describe UNilateral Self-Address Translation (UNSAF) mechanisms to establish TCP through NATs by modifying only endpoints. These approaches depend on specific NAT behavior that is not always supported (see [6] and [5] for details). Consequently a complete TCP NAT Traversal solution is sometimes forced to rely on public TCP Relays. This document defines requirements that ensures that TCP NAT Traversal approaches are not forced to use data relays.

3. 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 RFC 2119 [7].

This document uses the term "session" as defined in RFC 2663 [2]. "NAT" in this specification includes both "Basic NAT" and "Network Address/Port Translator (NAPT)" [2].

This document uses the terms "address and port mapping", "endpoint independent mapping", "filtering behavior", "endpoint independent filtering", "address dependent filtering" and "address and port dependent filtering" as defined in RFC FIXME-BEHAVE-UDP [1].

4. TCP Session Setup

This section describes various NAT behaviors applicable to TCP session setup.

4.1 Address and Port Mapping

RFC FIXME-BEHAVE-UDP [1] defines the criteria for the re-use of a mapping for new sessions. The definition presented there is agnostic of the transport protocol used and applies directly to TCP.

REQ-1: A NAT MUST have an "External NAT mapping is endpoint independent" behavior.

Justification: REQ-1 is necessary for UNSAF methods to work. Refer to REQ-1 in [1] for details.

4.2 Internally Initiated Sessions

An internal endpoint initiates a TCP session through a NAT by sending a SYN packet. The NAT assigns an external IP address and port number for the session so the resulting SYNACK response can be received, translated and routed to the internal endpoint. This translation is used for the subsequent ACK and other packets for the duration of the session. This corresponds to the 3-Way Handshake mode of session initiation defined in RFC 793 [8] and is supported by all NATs.

RFC 793 defines an alternate mode of session initiation, termed Simultaneous-Open, which is used by peer-to-peer applications to traverse NATs. In the Simultaneous-Open mode of operation, both endpoints send SYN packets that cross in the network, followed by SYNACK packets that cross in the network. From the perspective of the NAT, the internal host's SYN packet is responded by an inbound SYN packet for the same session (as opposed to a SYNACK packet). Subsequent to this exchange, both an outbound and inbound SYNACK are seen for the session. Some NATs block the inbound SYN for the session; some NATs block or incorrectly translate the outbound SYNACK. Such behavior breaks TCP Simultaneous-Open and prevents peer-to-peer applications from functioning correctly behind a NAT.

In order to provide network address translation service for TCP, it is necessary for a NAT to correctly receive, translate, and forward all packets for a session that conforms to valid transitions of the

TCP State-Machine [8].

REQ-2: For a TCP session, a NAT MUST support all valid sequences of TCP packets as defined in RFC 793. In particular:
a) A NAT MUST support TCP Simultaneous-Open.

Justification: This requirement enables standards compliant TCP stacks to traverse NATs.

4.3 Externally Initiated Sessions

When an internal endpoint initiates a session, the NAT assigns an external IP:port. Some peer-to-peer applications let other external endpoints initiate a TCP session to the internal endpoint by sending a SYN to the external IP:port allocated. Such applications depend on the NAT to reuse the mapping and route the SYN to the internal endpoint. The internal endpoint replies with a SYNACK packet that the NAT is expected to translate and forward to the external endpoint, which responds with an ACK to complete the initiation. The filtering behavior of the NAT, defined in [1], governs which external endpoints are allowed to send inbound SYN packets for a new session.

REQ-3: A NAT with "Endpoint independent filtering" or "Address dependent filtering" behavior MUST support TCP session initiations from the specific external endpoints. Note that this requirement is not applicable to NATs that have "Address and port dependent filtering" behavior.

Justification: This is to avoid breaking peer-to-peer applications which do not always initiate sessions from the internal side of the NAT.

If the inbound SYN packet is filtered, either because a corresponding mapping does not exist or because of the NAT's filtering behavior, a NAT has two basic choices: to ignore the packet silently, or signal an error to the sender. Ignoring the SYN helps applications perform TCP Simultaneous-Open in the presence of clock skew and network congestion where the inbound SYN may arrive at the NAT before the outbound SYN creates the necessary session state.

REQ-4: It is RECOMMENDED that a NAT silently discard inbound SYN packets that are filtered or cannot be routed.

Justification: This allows applications to traverse NATs with greater ease.

5. TCP Session Refresh

A NAT maintains state associated with new and established sessions. Because of this, a NAT is susceptible to a resource-exhaustion attack whereby an attacker (or virus) on the internal side attempts to cause the NAT to create more state than it has resources for. To prevent such an attack, a NAT needs to abandon sessions in order to free the state resources.

A common method that is applicable only to TCP sessions is to preferentially abandon sessions for crashed endpoints, followed by closed TCP sessions and partially-open sessions. A NAT can check if an endpoint has crashed by sending a TCP keep-alive packet and checking for the response. If the NAT cannot determine whether the session is active, it should not abandon it until the session has been idle for some time. The time is derived from values recommended in RFC 1122 [9]. The states of a TCP session that these values correspond to are defined in RFC 793 [8] and can be inferred by passively examining the TCP flags of inbound and outbound packets for that session.

The established session timer is defined as the time a mapping will stay active for a session over which application data can be exchanged. Application data can be exchanged over a TCP session in states: ESTABLISHED, FIN_WAIT_1, FIN_WAIT_2, and CLOSE_WAIT.

The transitory session timer is defined as the time a mapping will stay active for a session over which application data cannot yet be exchanged, or can no longer be exchanged. This includes partially-open TCP sessions in states: SYN_SENT and SYN_RCVD, and closed sessions in states: CLOSING, LAST_ACK, and TIME_WAIT.

REQ-5: If a NAT cannot determine whether the endpoints of a TCP session are active, it MAY abandon the session if it has been idle for some time. A default value of 2 hours for the established session timer is RECOMMENDED. A default value of 4 minutes for the transitory session timer is RECOMMENDED.
a) The value of the NAT TCP session timers MAY be configurable.

Justification: If a NAT cannot determine whether the endpoint of an idle TCP session has crashed, the NAT should assume that the endpoint is active. However, to defend against DoS attacks, a NAT can abandon session state under certain circumstances while minimally impacting active endpoints. For idle TCP sessions where data can be exchanged (that is, once ACK packets are seen in both directions, and FIN packets have not been seen in both directions), some endpoints send keep-alive packets at 2 hour intervals by default. For idle TCP sessions that are partially-

open or closed, TCP waits 2xMSL (4 minutes) for in-flight packets to be delivered and acknowledged. If a NAT passively waits for at least this interval and does not see any packets for the TCP session, it can prematurely abandon the session without impacting most applications. NAT behavior for handling RST packets for a session is left undefined.
a) Configuration helps troubleshoot and accommodate specific applications.

6. Application Level Gateways

FIXME OPEN ISSUE: Is this out-of-scope? Do we need to specify all TCP ALGs should be off by default? What about FTP?

7. Requirements

A NAT that supports all of the mandatory requirements of this specification (i.e., the "MUST") and is compliant with [1], is "compliant with this specification." A NAT that supports all of the requirements of this specification (i.e., included the "RECOMMENDED") and is fully compliant with [1] is "fully compliant with all the mandatory and recommended requirements of this specification."

REQ-1: A NAT MUST have an "External NAT mapping is endpoint independent" behavior.
REQ-2: For a TCP session, a NAT MUST support all valid sequences of TCP packets as defined in RFC 793. In particular:
a) A NAT MUST support TCP Simultaneous-Open.
REQ-3: A NAT with "Endpoint independent filtering" or "Address dependent filtering" behavior MUST support TCP session initiations from the specific external endpoints. Note that this requirement is not applicable to NATs that have "Address and port dependent filtering" behavior.
REQ-4: It is RECOMMENDED that a NAT silently discard inbound SYN packets that are filtered or cannot be routed.
REQ-5: If a NAT cannot determine whether the endpoints of a TCP session are active, it MAY abandon the session if it has been idle for some time. A default value of 2 hours for the established session timer is RECOMMENDED. A default value of 4 minutes for the transitory session timer is RECOMMENDED.
a) The value of the NAT TCP session timers MAY be configurable.

8. Security considerations

In addition to the security considerations addressed in [1], there are additional concerns for handling TCP packets and are discussed in this section.

Security considerations for REQ-1: This requirement does not introduce any TCP-specific concerns in addition to those already addressed in [1].

Security considerations for REQ-2: This document requires that a NAT accept an inbound SYN packet for a session in response to the outbound SYN packet. In order to provide extra security, some NATs require the ACK flag to be set in all inbound packets. This attempts to protect against attackers that can blindly spoof SYN packets appearing to come from the external destination, but are not able to receive, and therefore acknowledge, packets addressed to the same. REQ-2 in this document does not prevent a NAT from providing the same security guarantees. Even after the inbound SYN is accepted, the external endpoint is required by the TCP specification to explicitly acknowledge the internal endpoint's sequence number through a subsequent SYNACK or ACK packet. The NAT can check these subsequent packets to thwart such spoofing attacks.

Security considerations for REQ-3: The security provided by the NAT is governed by its filtering behavior as addressed in [1]. Allowing TCP sessions to be initiated by endpoints in accordance with the filtering behavior does not introduce additional concerns.

Security considerations for REQ-4: This document recommends that if an inbound SYN packet is filtered, then the NAT should silently discard it. Some NATs send TCP RST or ICMP errors in response to filtered packets. This serves to protect the NAT'ed hosts from identity-theft attacks. In such an attack, the NAT is the rightful recipient for an address, but an attacker blindly spoofs packets from this address. If the NAT silently drops unexpected inbound packets then the attacker can potentially spoof an entire TCP session and masquerade as the NAT'ed endpoint. REQ-4 allows a NAT to respond to such attacks by sending error packets for unexpected non-SYN packets that follow the SYN packet.

Security considerations for REQ-5: This document recommends that a NAT that passively monitors session state keep idle TCP sessions alive for at least 4 minutes for partially-open or closed sessions, and for at least 2 hours for established sessions by default. If a NAT is under a DoS attack, the NAT administrator may configure session timeouts accordingly, or let the NAT actively determine session state.

NAT implementations that change local state based on TCP flags in packets must ensure that out-of-window TCP packets are properly handled. Out-of-window TCP packets are sometimes used in attacks where an attacker resets arbitrary TCP sessions by guessing only the endpoint IP addresses and ports. If the window is too large, an attacker can send a small number of packets with crafted sequence numbers such that one of these packets is considered an in-window

packet that resets the session.

9. IANA considerations

This document does not change or create any IANA-registered values.

10. Acknowledgments

Thanks to Paul Hoffman for his many contributions to this document.

11. Normative References

[1] Audet, F. and C. Jennings, "NAT Behavioral Requirements for Unicast UDP", draft-ietf-behave-nat-udp (work in progress).

[2] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[3] Guha, S. and P. Francis, "NUTSS: A SIP based approach to UDP and TCP connectivity", Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (Portland, OR), August 2004.

[4] Biggadike, A., Ferullo, D., Wilson, G., and A. Perrig, "NATBLASTER: Establishing TCP connections between hosts behind NATs", Proceedings of the ACM SIGCOMM Asia Workshop (Beijing, China), April 2005.

[5] Ford, B., Srisuresh, P., and D. Kegel, "Peer-to-peer communication across network address translators", Proceedings of the USENIX Annual Technical Conference (Anaheim, CA), April 2005.

[6] Guha, S. and P. Francis, "Characterization and Measurement of TCP Traversal through NATs and Firewalls", Proceedings of the Internet Measurement Conference (Berkeley, CA), October 2005.

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

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

[9] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

Authors' Addresses

Saikat Guha
Cornell University
331 Upson Hall
Ithaca, NY 14853
US

Email: saikat@cs.cornell.edu

Kaushik Biswas
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134
US

Phone: +1 408 525 5134
Email: kbiswas@cisco.com

Bryan Ford
M.I.T.
Laboratory for Computer Science
77 Massachusetts Ave.
Cambridge, MA 02139
US

Phone: +1 617 253 5261
Email: baford@mit.edu

Paul Francis
Cornell University
4108 Upson Hall
Ithaca, NY 14853
US

Phone: +1 607 255 9223
Email: francis@cs.cornell.edu

Senthil Sivakumar
Cisco Systems, Inc.
7100-8 Kit Creek Road
PO Box 14987
Research Triangle Park, NC 27709-4987
US

Phone: +1 919 392 5158
Email: ssenthil@cisco.com

Pyda Srisuresh
Consultant
20072 Pacifica Dr.
Cupertino, CA 95014
US

Phone: +1 408 836 4773
Email: srisuresh@yahoo.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.