

BEHAVE WG
Internet-Draft
Intended status: BCP
Expires: May 8, 2009

F. Gont
UTN/FRH
P. Srisuresh
Kazeon Systems, Inc.
November 4, 2008

Security implications of Network Address Translators (NATs)
draft-gont-behave-nat-security-01.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79. This document may not be modified, and derivative works of it may not be created.

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/lid-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 May 8, 2009.

Abstract

This document analyzes the security implications of Network Address Translators (NATs). It neither deprecates nor encourages the use of NATs, but rather aims to raise awareness about their security implications, and possible implementation approaches to improve their security.

Table of Contents

- 1. Introduction 3
- 2. Security Implications from IP fragmentation 3
 - 2.1. Fragment processing for inbound IP packets 3
 - 2.2. Fragment processing on the outbound 4
- 3. Port reservation 5
- 4. DHCP-Configured NATs 5
 - 4.1. DHCP-Configured NATs in a Multi-Level NAT deployments . . 6
 - 4.2. DHCP-Configured NATs in a Remote Access VPN operation . . 6
- 5. Security considerations arising from protocol header fields . 6
 - 5.1. Internet Protocol version 4 (IPv4) header fields 6
 - 5.1.1. Version 6
 - 5.1.2. IHL 6
 - 5.1.3. Type of Service 7
 - 5.1.4. Total Length 7
 - 5.1.5. Identification 7
 - 5.1.6. Flags 7
 - 5.1.7. Fragment Offset 7
 - 5.1.8. Time to Live 7
 - 5.1.9. Protocol 8
 - 5.1.10. Header Checksum 8
 - 5.1.11. Source Address 8
 - 5.1.12. Destination Address 8
 - 5.1.13. Options 8
 - 5.1.14. Padding 8
 - 5.2. Transmission Control Protocol (TCP) header fields 8
 - 5.2.1. Source Port 8
 - 5.2.2. Destination Port 9
 - 5.2.3. Sequence Number 9
 - 5.2.4. Acknowledgment Number 9
 - 5.2.5. Data Offset 10
 - 5.2.6. Reserved 10
 - 5.2.7. Flags 10
 - 5.2.8. Window 10
 - 5.2.9. Checksum 10
 - 5.2.10. Urgent Pointer 10
 - 5.2.11. Options 10
 - 5.2.12. Padding 11
- 6. Security Considerations 11
- 7. IANA Considerations 11
- 8. Acknowledgements 11
- 9. References 11
 - 9.1. Normative References 11
 - 9.2. Informative References 12
- Authors' Addresses 13
- Intellectual Property and Copyright Statements 14

1. Introduction

This document analyzes the security implications of Network Address Translators (NATs). It neither deprecates nor encourages the use of NATs, but rather aims to raise awareness about their security implications, and possible implementation approaches to improve their security.

Section 2 and Section 3 analyze the possible security implications of basic NAT functionality. Section 4 analyzes the possible security implications of DHCP-Configured NATs. Section 5 analyzes the possible security implications arising from the non-modification of protocol header fields by NATs.

Note: the security implications of a NAT device due to it being a stateful device are not discussed in the current version of this document (but may be added in future revisions). For what is worth, many of these security implications are described in [RFC5382], [RFC4787] and [I-D.ietf-behave-nat-icmp].

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].

2. Security Implications from IP fragmentation

2.1. Fragment processing for inbound IP packets

Routers in the network are able to forward fragmented IP packets just as they do any other non-fragmented IP packets because packet forwarding is based solely on looking up the destination IP address in the routing table and finding the largest prefix match to identify the next-hop to forward to. Routers do not need to retain any state pertaining to fragmented packets traversing them.

A NAT device operates differently from a router in that the NAT device must find the matching NAT Session for an IP packet and perform NAT translation on the packet, prior to forwarding. NAT Session lookup requires the full 5-tuple of the IP datagram. Only the first fragment of the IP datagram contains the full-tuple. Subsequent fragmented packets contain the fragment Id, but do not contain transport protocol specific details such as source and destination port numbers. The NAT device must be able to associate the same session tuple for all fragments by virtue of the fragment ID and use that information to locate the NAT Session the packets belong to. Note however, the IP fragments cannot be assumed to arrive in order. Some operating systems transmit the fragments of an IP

datagram out of their logical order as a matter of course. In addition, network conditions can also cause dynamic packet reordering in transit.

A NAT device not capable of processing all fragments of an inbound IP datagram can cause the fragmented packets to be dropped causing some applications to not function correctly.

NATs, capable of processing out-of-order packets store the out-of-order packets prior to forwarding. This can open up the NAT device for external attacks. As pointed out in [RFC4787], fragmentation has been a tool used in many attacks, some involving passing fragmented packets through NATs, and others involving DoS attacks based on the state needed to reassemble the fragments. NAT implementers should be aware of [RFC3128] and [RFC1858].

NATs may protect themselves against such attacks by limiting the length of time they retain an incomplete IP packet before discarding it, or by limiting the amount of internal buffer space incomplete IP packets may consume before the oldest fragments are discarded. The appropriate values of these limits vary across NATs, and may be determined by the network administrator.

[CPNI-IP] contains a detailed discussion of the security implications arising from the reassembly of IP fragments and of a number of approaches to mitigate them.

2.2. Fragment processing on the outbound

Say, two private hosts originated TCP/UDP packets (fragmented or not) to the same destination host and both packets transit the same NAT device and use the same fragmentation identifier. Say, the NAT device assembled the IP packets (in the case they were fragmented) and translated the same using the appropriate NAT Sessions. When NAT translates IP datagrams, it would assign all outbound IP datagrams the same Public IP, but different TCP/UDP numbers. While forwarding, an IP datagram may be fragmented on the way out. Only the first fragment contains the TCP/UDP header that would be necessary to associate the packet to a specific session. Subsequent fragments do not contain TCP/UDP port information, but simply carry the same fragmentation identifier specified in the first fragment.

When the target host receives the two unrelated datagrams, carrying same fragmentation id, and from the same assigned host address, the target host is unable to determine which of the two sessions the datagrams belong to and might corrupt both sessions. This can be a security breach any of the sessions associated.

In order to avoid problems of this kind, the NAPT device SHOULD further translate fragment ID in the outgoing packets such that the tuples of (SrcIP, DestIP, fragment Id, Protocol) are unique and distinguishable across all outgoing packets from the NAT device.

When fragmenting packets on the outbound, a NAT device implementing NAPT function SHOULD ensure that the tuples of (SrcIP, DestIP, fragment Id, Protocol) are unique across all outgoing packets.

3. Port reservation

A NAPT device often shares the source ports for its public IP address with nodes in the private realm. Reserving port blocks explicitly for local use vs. NAT use is valuable for several reasons. Consider the following scenario.

Say, an application on the NAPT runs on port 5060 (SIP Server), but not enabled. A host in the private domain uses 5060 at this time and say, gets the port 5060 from the NAT device. While this Port Binding is active, say, the application running on NAT is activated. Several things can go wrong now depending on the implementation:

1. The application is totally unaware of NAT's existence, (maybe because NAT never does a bind on the ports it is using). So it starts using 5060 and the subsequent packets directed to this server could end up in the end host within the private domain or the packets meant for the application on the end host could end up in the NAT box's TCP/UDP stack. Both are bad and can cause unpredictable behavior.
2. The application on the NAT box is aware that someone is using 5060 so the Bind fails and the app fails to come up. The administrator would have to clear the NAT session and restart the application.
3. The application on the NAT box is intelligent enough to tell NAT to clean up any sessions that it plans to use and NAT cleans up its session(s). The application on the end host is effected as a result.

Clearly, there can be unpredictable behavior when ports are not reserved explicitly for local use vs NAT use.

4. DHCP-Configured NATs

4.1. DHCP-Configured NATs in a Multi-Level NAT deployments

Many NATs, particularly consumer-level devices designed to be deployed by nontechnical users, also act as DHCP [RFC2131] clients. In its default configuration, a consumer NAT typically obtains its public IP address, default router, and other IP configuration information via DHCP from an ISP or other "upstream" network.

On its internal network side, the NAT then automatically sets up its own private "downstream" subnet in one of the private IP address regions assigned to this purpose in [RFC1918]. The NAT typically acts as a DHCP server for its private downstream network, managing its pool of private IP addresses automatically and handing them out to the hosts (and perhaps other NATs) on the private network on demand.

This auto-configuration of private networks can be problematic, if the NAT's upstream network is also in RFC 1918 private address space. In the Multi Level NAT deployments, end-hosts could have their security compromised due to mistaken server identities as described in section 3 of [I-D.ford-behave-top].

4.2. DHCP-Configured NATs in a Remote Access VPN operation

In deployments where a corporate network deploys the same private address space as used on sundry remote locations, end-hosts could have their security compromised due to mistaken server identities, as described in section 4 of [I-D.ford-behave-top].

5. Security considerations arising from protocol header fields

The following subsections analyze the security implications arising from arising from the non-modification of protocol header fields by Network Address Translators (NATs)

5.1. Internet Protocol version 4 (IPv4) header fields

5.1.1. Version

This field does not require any special handling by NATs.

5.1.2. IHL

This field does not require any special handling by NATs.

5.1.3. Type of Service

End-systems have traditionally selected different TOS values depending on the nature of the application making use of the IP services. As the TOS value selected for each packet usually depends the specific IP implementation, this could be exploited to roughly count the number of systems behind a NAT. In order to avoid the TOS field from revealing this information, NATs could rewrite the TOS field in outgoing packets according to the Protocol value in the IP header, and the Destination Port value in the header of the transport protocol running on top of IP.

5.1.4. Total Length

This field does not require any special handling by NATs.

5.1.5. Identification

The IP Identification field is used for the reassembly of IP fragments. Most IP implementations have typically selected the IP Identification field from a global counter that is incremented by one each time a packet is transmitted [CPNI-IP]. As discussed in [Bellovin2002], the Identification field can be exploited to count the number of systems behind a NAT, thus unnecessarily revealing information about the "internal" network. In order to avoid this issue, NATs could rewrite the IP Identification field such that it is not trivial for an attacker to detect different "sequences" of the Identification field. [CPNI-IP] discussed a number of approaches for selecting the Identification value at end-systems, which could also be applied for the selection of the Identification value at NATs.

It should be noted that if a NAT does not rewrite the Identification field, a given Identification value could end up being reused too quickly, with the potential of interoperability problems.

5.1.6. Flags

5.1.7. Fragment Offset

This field does not require any special handling by NATs.

5.1.8. Time to Live

As discussed in [CPNI-IP], the IP Time to Live field can be exploited to:

- o Fingerprinting the operating system being used by the source host.
- o Fingerprinting the physical device from which the packets originate.
- o Locating the source host in the network topology.

In order to avoid having the IP Time to Live field reveal this information, NATs could rewrite the TTL field of translated packets, such that this field is set homogeneously among all packets forwarded toward the external network.

5.1.9. Protocol

This field does not require any special handling by NATs.

5.1.10. Header Checksum

This field does not require any special handling by NATs.

5.1.11. Source Address

Here we make no special considerations about this field.

5.1.12. Destination Address

Here we make no special considerations about this field

5.1.13. Options

Clearly, IP options could potentially be used for counting the number of systems behind a NAT. However, as it is unusual for end-systems to include IP options in the IP packets they send, in most cases this IP options will not require any special handling by NATs.

5.1.14. Padding

This field does not require any special handling by NATs.

5.2. Transmission Control Protocol (TCP) header fields

5.2.1. Source Port

As part of its basic functionality, a NAT-PT will usually rewrite (translate) the TCP Source Port of packets sent to the external realm. As a result, the ephemeral port selection algorithm of a NAT will "override" that of the end-systems behind the NAT.

In some cases, this may have the undesirable consequence that a system implementing some algorithm for ephemeral port obfuscation may end up establishing TCP connections with systems in the external realm using a predictable (as seen from the external realm) ephemeral port sequence.

NATs should implement an ephemeral port selection algorithm such that the source port of outgoing packets is obfuscated, thus mitigating blind (off-path) spoofing attacks.

It should be noted that use of an improper ephemeral port selection algorithm may lead to collisions of connection-ids, with the potential of failure in the establishment of new TCP connections. [I-D.ietf-tsvwg-port-randomization]

5.2.2. Destination Port

Here we make no special considerations for this field.

5.2.3. Sequence Number

The choice of the Initial Sequence Number of a connection by an end-system is not arbitrary, but aims to minimize the chances of a stale segment from being accepted by a new incarnation of a previous connection. [RFC0793] suggests the use of a global 32-bit ISN generator, whose lower bit is incremented roughly every 4 microseconds. Based on the premise that the ISNs of consecutive TCP connections are monotonically-increasing, BSD-derived implementations use the ISN of an incoming connection request to perform heuristics aiming at allowing a new incarnation of a previous connection to be created, even if the previous incarnation is still in the TIME-WAIT state. However, an ISN such as that described in [RFC0793] makes it trivial for an attacker to predict the ISN used for future connections, thus making it easier for the attacker to perform "blind" attacks against those connections.

NATs should rewrite the Sequence Number of outgoing segments such that consecutive connections to a specific service at a specific system use ISNs that are monotonically-increasing. Additionally, the ISN generator should be such that it should be difficult for an off-path attacker to predict the ISNs of future connections. [RFC1948] describes an algorithm for the generation of ISN that complies with these two "requirements".

5.2.4. Acknowledgment Number

Here we make no special considerations for this field.

5.2.5. Data Offset

Here we make no special considerations for this field.

5.2.6. Reserved

Here we make no special considerations for this field.

5.2.7. Flags

Here we make no special considerations for this field.

5.2.8. Window

Here we make no special considerations for this field.

5.2.9. Checksum

Here we make no special considerations for this field.

5.2.10. Urgent Pointer

Here we make no special considerations for this field.

5.2.11. Options

5.2.11.1. TCP timestamps

The Timestamps option, specified in RFC 1323 [RFC1323], allows a TCP to include a timestamp value in its segments, that can be used to perform two functions: Round-Trip Time Measurement (RTTM), and Protect Against Wrapped Sequences (PAWS).

For the purpose of PAWS, the timestamps sent on a connection are required to be monotonically increasing. While there is no requirement that timestamps are monotonically increasing across TCP connections, the generation of timestamps such that they are monotonically increasing across connections between the same two endpoints allows the use of timestamps for improving the handling of SYN segments that are received while the corresponding four-tuple is in the TIME-WAIT state. That is, the timestamp option could be used to perform heuristics to determine whether to allow the creation of a new incarnation of a connection that is in the TIME-WAIT state.

NATs could rewrite the TCP Timestamps option such that TCP consecutive connections with a specific service at a specific system use monotonically increasing timestamps (i.e., the TCP timestamps are monotonically-increasing across those

connections). [I-D.gont-tcpm-tcp-timestamps] describes an algorithm that complies with this requirement.

5.2.12. Padding

Here we make no special considerations for this field.

6. Security Considerations

This document analyzes the security implications of Network Address Translators (NATs). It neither deprecates nor encourages the use of NATs, but rather aims to raise awareness about their security implications, and possible implementation approaches to improve their security.

Note: the security implications of a NAT device due to it being a stateful device are not discussed in the current version of this document (but may be added in future revisions). For what is worth, many of these security implications are described in [RFC5382], [RFC4787] and [I-D.ietf-behave-nat-icmp].

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

9. References

9.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.
- [RFC1323] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

- [Bellovin2002]
Bellovin, S., "A Technique for Counting NATted Hosts",
IMW'02 Nov. 6-8, 2002, Marseille, France, 2002.
- [CPNI-IP] CPNI, "Security Assessment of the Internet Protocol",
2008 .
- [CPNI-TCP]
CPNI, "Security Assessment of the Transmission Control
Protocol (TCP)", (to be published) .
- [I-D.ford-behave-top]
Srisuresh, P. and B. Ford, "Complications from Network
Address Translator Deployment Topologies",
draft-ford-behave-top-04 (work in progress), October 2008.
- [I-D.gont-tcpm-tcp-timestamps]
Gont, F., "On the generation of TCP timestamps",
draft-gont-tcpm-tcp-timestamps-00 (work in progress),
October 2008.
- [I-D.ietf-behave-nat-icmp]
Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT
Behavioral Requirements for ICMP protocol",
draft-ietf-behave-nat-icmp-10 (work in progress),
October 2008.
- [I-D.ietf-tsvwg-port-randomization]
Larsen, M. and F. Gont, "Port Randomization",
draft-ietf-tsvwg-port-randomization-02 (work in progress),
August 2008.
- [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security
Considerations for IP Fragment Filtering", RFC 1858,
October 1995.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and
E. Lear, "Address Allocation for Private Internets",
BCP 5, RFC 1918, February 1996.
- [RFC1948] Bellovin, S., "Defending Against Sequence Number Attacks",
RFC 1948, May 1996.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol",

RFC 2131, March 1997.

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, June 2001.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC5128] Srisuresh, P., Ford, B., and D. Kegel, "State of Peer-to-Peer (P2P) Communication across Network Address Translators (NATs)", RFC 5128, March 2008.
- [RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

Authors' Addresses

Fernando Gont
Universidad Tecnologica Nacional / Facultad Regional Haedo
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fernando@gont.com.ar
URI: <http://www.gont.com.ar>

Pyda Srisuresh
Kazeon Systems, Inc.
1161 San Antonio Rd.
Mountain View, CA 94043
U.S.A.

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

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

