BEHAVE WG
Internet-Draft
Intended status: BCP
Expires: April 29, 2010

F. Gont
UTN/FRH
P. Srisuresh
EMC Corporation

October 26, 2009

Security implications of Network Address Translators (NATs)
draft-gont-behave-nat-security-03.txt

Status of this Memo

This Internet-Draft will expire on April 29, 2010.

Copyright Notice

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

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].

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. Resilience to Denial of Service (DoS) attacks

2.1. IP fragmentation attacks

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.

REQ-1: A NAT device capable of forwarding out-of-order IP fragments MUST take measures to protect itself against well-known IP fragment based attacks.

3. Port reservation

A NAT device implementing NAPT function shares the source ports for its public IP address with nodes in the private realm. The NAPT device may also have end host applications of its own. Consider the following scenario when a NAPT device uses the same TCP/UDP port for local use as well as for mapping to a private host.

Say, an application on the NAPT device runs on port 5060 (SIP Server), but not enabled. Say, a host in the private domain of the Nat device also uses 5060 and obtains port 5060 from the NAT device. While this Port Binding is active, say, the application on NAPT device is activated. The application on the NAT device is unlikely to be aware of the NAT function enforced by the NAT device. Once the same port is assigned for NAT use as well as for use by local application on the NAT device, data packets directed to the NAT device could end up with the end host within the private domain or the application on the NAT device. This behavior can cause unpredictable behavior and may even result in data snooping.

Manual intervention becomes necessary to ensure that only one instance of an application is actively using a port at a given time.

It is not desirable to either allow possible simulataneous use (or) require manual intervention to serialize the use.

REQ-2: When a NAT device supports local applications on the device, it is RECOMMENDED that the NAT device reserves specific ports for local use, different from NAT use, so there is no overlap of ports between local use and NAT use. Doing this will ensure there is no possibility of cross session contamination between NAT sessions and local sessions.

4. Peer-to-peer Communication for the end hosts behind devices

[RFC5128] refers to applications using TCP/UDP hole punching technique to establish peer-to-peer communication. Section 6.1 of [RFC5128] describes a scenario in which it can be problematic for applications that do not use appropriate authentication mechanisms while setting up peer connections. An application could end up connecting to the wrong host or have its connections hijacked maliciously by other hosts.

REQ-3: Applications attempting to establish peer-to-peer communication across NAT devices using TCP/UDP hole punching technique SHOULD employ relevant authentication mechanism to connect to their peers.

5. Secure Transport for the end hosts behind NAT Devices

NAT devices are ubiquitous in the Internet. NAT devices can be found in homes, hotels, Airports, conferences, coffee shops and plethora of internet cafes.

Most users needing to carry out financial transactions and other personal, sensitive applications use SSL/TLS protocol [RFC2246] to do this. NAT devices enroute MUST support the traversal of SSL/TLS protocol. TCP port 443 is the default port used for SSL/TLS protocol.

Secure VPNs is another important use of secure protocols to access corporate networks. Telecommuters and users in remote locations use secure VPN to access their corporate networks. The secure VPNs may use a combination of NAT-T [RFC3947] and IPSec-over-UDP [RFC3948] to secure the VPN traffic. Alternately, some VPN vendors use SSL/TLS protocol [RFC2246] to secure the VPN traffic. NAT devices enroute MUST support the traversal of NAT compliant security protocols such as SSL/TLS, NAT-T and IPsec-over-UDP.

Enforcement of NAT-T for IKE negotiation can be problematic, as described in Section 2.3 of [RFC3715], if the NAT device enroute has an Application Level Gateway (ALG) that attempts to treat IKE packets differently from other UDP packets. A NAT device MUST NOT include an ALG that treats IKE or SSL/TLS packets differently than any other TCP/UDP packet.

REQ-4: A NAT device MUST permit the traversal of NAT compliant security protocols. Specifically, a NAT device MUST do the following.

a. A NAT device MUST NOT block traffic directed to or coming from UDP port numbers 500 and 4500.

b. A NAT device MUST NOT block traffic directed to or coming from TCP/UDP port number 443.

c. A NAT device MUST NOT include an ALG that treats IKE packets or SSL/TLS packets differently than any other TCP/UDP packet.

6. Security considerations arising from protocol header fields

From the external realm, all packets originated by any host in the internal realm (or by the NAT itself) are seen as originating from the NAT box. As a result, the same requirements that are applied to an internet host must be applied to the aggregate traffic at the NAT box. For example, in the same way that an internet host is required not to reuse a tuple (SrcIP, DstIP, Protocol, IP ID) at any given time, a NAT should enforce that a tuple (SrcIP, DstIP, Protocol, IP ID) of the aggregate traffic is not reused at any given time.

In order to enforce some of these requirements, a NAT will usually need to rewrite some of the TCP and/or IP header fields of the incoming and outgoing packets.

In some cases, rewriting a header field can be mandatory to ensure interoperability (e.g., the IP Identification field). In other cases, rewriting a header field (e.g., the TCP Sequence Number) ensures that the NAT will not introduce new interoperability problems in some corner cases

If a NAT implementation opts to rewrite some header field, there is still the question of how to do it security-wise.

The following subsections discusses those header fields that may need to be rewriten by the NAT to avoid interoperability problems, and discusses the best possible policies to rewrite them.

6.1. Internet Protocol version 4 (IPv4) header fields

6.1.1. Identification

For interoperability reasons, NATs must ensure that a tuple (SrcIP, DstIP, Protocol, IP ID) is not reused while there are still packets in the network with that tuple. In order to enforce this requirement, NATs MUST rewrite the IP Identification field of the outgoing IP packets.

A trivial approach to enforce this requirement would be to rewrite the IP Identification from a global counter that is increment by one each time a packet is transmitted. However, while this would fulfil the interoperability requirements, this would lead to predictable Identification values, which have been found to have a number of security implications [CPNI-IP].

In order to mitigate these security implications, NATs SHOULD rewrite the IP Identification field such that it is not trivial for an attacker to detect different "sequences" of the Identification field. [CPNI-IP] discusses 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.

REQ-5: NATs MUST ensure that a tuple (SrcIP, DstIP, Protocol, IP ID) is not reused while there are still packets in the network with that tuple. Additionally, they SHOULD generate the IP Identification values such that they are not trivially predictable.

6.2. Transmission Control Protocol (TCP) header fields

6.2.1. Source Port

As part of its basic functionality, a NAPT 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 undesireable 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]

6.2.2. Sequence Number

Based on the premise that the Initial Sequence Numbers (ISNs) of succesive TCP connections are monotonically-increasing, BSD-derived implementations use the ISN of an incomming connection request to perform heuristics aiming at allowing a new incarnation of a previous connection to be created, even if the pervious incarnation is still in the TIME-WAIT state.

When succesive TCP connection requests are sent from different nodes in the internal realm to a node in the external realm, the resulting ISN sequence may not be monotonically-increasing (even if if every node in the external real enforces monotonically-increasing ISNs for their own connection requests). As a result, succesive connection requests through a NAT may result in a connection reset or may simply time-out.

One possible workaround for this problem would be to maintain a TIME-WAIT state (for 2*MSL seconds) for every connection that is closed, so that a given four-tuple is not reused two quickly. However, this increases the state that must be kept at the NAT (every terminated TCP connection requires the NAT to maintain state for an additional 2*MSL seconds), and might also reduce the maximum connection-establishment rate through the NAT.

An alternative workaround would be to have the NAT rewrite the Sequence Number of outgoing segments such that consecutive connections to a specific TCP endpoint use ISNs that are monotonically-increasing. From a security point of view, 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".

REQ-6: A NAT MAY rewrite the TCP Sequence Number of packets forwarded to the external realm, such that all connection requests from to a TCP endpoint in the external realm result in monotonically-increasing Initial Sequence Numbers (ISNs). The ISN generator SHOULD select Initial Sequence Numbers such that it is difficult for an off-path attacker to predict the ISNs of future connections. If the NAT rewrites the Sequence Number of packets forwarded to the external realm, it MUST also rewrite the TCP Acknowledgement Number of packets being forwarded into the internal realm.

6.2.3. Acknowledgment Number

As mentioned in Section 6.2.2, if the NAT rewrites the Sequence Number of TCP segments forwarded from the internal realm to the external realm, it must also rewrite the Acknowledgement Number of TCP segments forwarded from the external realm to the internal realm.

6.2.4. Options

6.2.4.1. 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 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, a number of systems for improving the handling of SYN segments that are received while the corresponding four-tuple is in the TIME-WAIT state, similar to the processing of TCP ISNs for connections in the TIME-WAIT state by BSD-derived systems. That is, the timestamp option is to perform heuristics to determine whether to allow the creation of a new incarnation of a connection that is in the TIME-WAIT state.

When succesive TCP connection requests are sent from different nodes in the internal realm to a node in the external realm, the resulting initial timestamps may not be monotonically-increasing (even if if every node in the external realm enforces monotonically-increasing timestamps across connection requests to the same destination endpoint). As a result, succesive connection requests through a NAT might result in a connection reset or might simply time-out.

One possible workaround for this problem would be to maintain a TIME-WAIT state (for 2*MSL seconds) for every connection that is closed, so that a given four-tuple is not reused two quickly. However, this increases the state that must be kept at the NAT (every terminated TCP connection requires the NAT to maintain state for an additional 2*MSL seconds), and might also reduce the maximum connection-establishment rate through the NAT.

An alternative workaround would be to have the NAT rewrite the TCP timestamp option of outgoing segments (TSval) such that consecutive connections to a specific TCP endpoint use timestamps that are monotonically-increasing. From a security point of view, the timestamps generator should be such that it makes it difficult for an

off-path attacker to predict the timestamps of future connections. [I-D.gont-tcpm-tcp-timestamps] describes an algorithm for the generation of timestamps that complies with these two "requirements". If the NAT rewrites the TCP timestamp of packets forwarded to the external realm, it must also rewrite the TCP timestamp echo (TSecr) of packets forwarded from the external realm into the internal realm.

REQ-7: A NAT MAY rewrite the TCP timestamps option(TSval) of packets forwarded to the external realm, such that all connection requests from to a specific TCP endpoint in the external realm result in monotonically-increasing timestamps. The timestamps generator SHOULD be such such that it makes it difficult for an off-path attacker to predict the timestamps of future connections. If the NAT rewrites the TCP timestamp of packets forwarded to the external realm, it MUST also rewrite the TCP timestamp echo (TSecr) of packets forwarded from the external realm into the internal realm.

7. 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].

8. IANA Considerations

This document has no actions for IANA.

9. Acknowledgements

The authors of this document would like to thank (in alphabetical order) Brian Carpenter, Remi Denis-Courmont, Reinaldo Penno, Dave Thaler, and Dan Wing for providing valuable feedback on earlier versions of this document.

10. References

10.1. 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.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC3715] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

10.2. 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, "Unintended Consequence of two NAT deployments with Overlapping Address Space", draft-ford-behave-top-07 (work in progress), August 2009.

[I-D.gont-tcpm-tcp-timestamps]
Gont, F., "On the generation of TCP timestamps", draft-gont-tcpm-tcp-timestamps-02 (work in progress), September 2009.

[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-12 (work in progress), January 2009.

[I-D.ietf-tsvwg-port-randomization]
Larsen, M. and F. Gont, "Port Randomization", draft-ietf-tsvwg-port-randomization-04 (work in progress), July 2009.

[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.

Appendix A. Change log (to be removed before publication of the document as an RFC)

A.1. Changes from draft-gont-behave-nat-security-01

- A number of sections were removed and/or reorganized.
- Where appropriate, requirements have been explicitly indicated as REQ-n.
- Updated Pyda's affiliation.
- Addressed part of the feedback received off-list from Reinaldo Penno.

A.2. Changes from draft-gont-behave-nat-security-01

- Addded Section 5.

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
EMC Corporation
1161 San Antonio Rd.
Mountain View, CA 94043
U.S.A.

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