INTERNET DRAFT
draft-ohta-e2e-nat-00.txt
Intended status: Informational
Expires: January 1, 2010

M. Ohta
Tokyo Institute of Technology

July 13, 2009

End to End NAT

Status of this Memo

Copyright Notice

Abstract

According to the end to end argument, NAT function can completely and correctly be implemented only with the knowledge and help of end hosts. By making NAT visible to the end hosts of NAT clients and let the hosts help NAT gateways, NAT actually becomes correct, complete, and end to end transparent. End to end NAT is upper compatible to legacy NAT while enabling various transport protocols (ICMP, SCTP, IPSEC), DNS reverse look up, Multicast and Mobile IP.

1. Introduction

NAT (Network Address Translation) is a technique to suppress address space consumption by sharing an IP [IP] address with multiple hosts at different times or, more practically, at the same time at different port numbers.

According to the end to end argument [SALTZER]:

> The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

NAT can completely and correctly be implemented only with the knowledge and help of the end hosts behind a NAT gateway.

However, legacy NAT was designed to try to make the NAT gateway transparent to the end hosts that the hosts can not help NAT gateways. As a result, legacy NAT is incomplete and incorrect in various ways, details of which is not discussed in this memo.

E2ENAT (End to end NAT) is a NAT, configuration of which is visible to the end hosts so that the hosts can, with their knowledge, help NAT gateways by reversing address translation at the hosts, limiting range of port used by the hosts and maintain state of NAT gateways, which makes NAT function complete and correct.

2. Operation of End to End NAT

2.1 Configuration of End to End NAT

Unless otherwise noted, the following configuration is assumed.

A NAT gateway has a public and a private interfaces. The public interface is connected to the Internet and the private one to a private network.

A shared public address is assigned to the NAT gateway.

Private addresses are addresses used by the private network. All other addresses are public addresses.

It is assumed that the end hosts in the private network knows NAT information, such as the shared public address and port numbers

allocated to each host, necessary to help NAT function through some mechanism such as DHCP, PPP or UPnP, details on which is not discussed in this memo.

2.2 Basic Operation of End to End NAT

NAT gateway, as usual, translate a destination addresses of a packet, if the destination address is the shared public address (after reassembly, if the packet is fragmented), based on a destination port number of the packet. TTL and IP check sum should also be modified appropriately.

To make end hosts sharing the shared public address, translation is applied to packets coming from both public and private interfaces of a NAT gateway.

However, port numbers and transport checksum must not be modified.

An end host translates a destination addresses of a packet back to the shared public address, if a source address of the packet is public. IP check sum should also be modified appropriately. Then, correctness of transport checksum should be automatically restored.

A source address of a packet from an end host with a private destination address should be private.

A source address of a packet from an end host with a public destination address should be the shared public address.

A source port number of a packet from an end host with a public destination address should be a port number assigned to the end host. So, there should be no port number collision which makes port number translation unnecessary.

An end host should output packet with a destination address of the shared public address, unless a destination port number is assigned to the host.

In a private network, a NAT gateway should advertise route to all the public addresses including the shared public one. Hosts sharing the address can communicate with the address through the NAT gateway.

A NAT gateway may assign some port numbers of the shared public address to itself and behave as an end host.

To support ICMP [ICMP] with a private source address generated against a packet from an end host, a NAT gateway should also translate destination address of an ICMP packet, if the ICMP packet

contains an internal packet source address of which is the shared public address, based on a source port number of the internal packet.

To ease management, a port number assigned to an end host should be valid for all the protocols of the host.

NAT gateways may be nested. That is, a public interface of an internal NAT gateway may be connected to a private network of an external NAT gateway. Port numbers allocated by the external NAT gateway to the internal NAT gateway will be further divided to end hosts in another private network behind the internal NAT gateway. Private addresses for the external gateway is public for the internal gateway.

2.3 Static and Dynamic End to End NAT

Depending on how port numbers are shared, there are static and dynamic E2ENAT or combinations of them. With static E2ENAT, an end host is assigned port numbers statically, which is necessary for a server with a stable IP address and a port number. With dynamic E2ENAT, end hosts dynamically share port numbers, which is useful if a small number of port numbers are shared by many end hosts, which could be the case with nested E2ENAT.

For dynamic E2ENAT, a NAT gateway and end hosts must somehow communicate, details of which is not discussed in this memo. However, it should be noted that connection state can and must be managed with the knowledge and help of end hosts. That is, end hosts should periodically refresh port number assignment so that no guessing of connection time out by a NAT gateway is necessary. Moreover, it is possible to have multiple NAT gateways sharing the shared public address, because states of the gateways can be synchronized completely and correctly by the end hosts.

2.4 Operating Servers under End to End NAT

End hosts behind NAT gateways can operate as servers with a stable address and a stable port numbers if, for example, a stable shared public and a stable private addresses and stable port numbers are assigned to the hosts.

Unlike port forwarding, E2ENAT offers full end to end transparency. That is, transport protocol other than TCP and UDP may be used and IP addresses may freely be contained in payload.

A server port number different from well known ones may be specified through mechanisms to specify an address of the server, which is the case of URLs. However, port numbers for DNS and SMTP are, in general,

implicitly assumed by DNS and are not changeable. When an ISP operate a NAT gateway, the ISP should, for fairness between customers, reserve some well know port numbers and assign small port numbers evenly to all the customers.

Or, a NAT gateway may receive packets to certain ports and behave as an application gateway to end hosts, if request messages to the server contains information, such as domain names, which is the case with DNS, SMTP and HTTP, to demultiplex the request messages to end hosts. However, for an ISP operating the NAT gateway, it may be easier to operate independent servers at default port for DNS, SMTP, HTTP and other applications for their customers than operating application relays.

2.5 Backward Compatibility of End to End NAT

The end to end principle requires end hosts of E2ENAT modified to support E2ENAT, if the hosts want to end to end transparency.

Instead, for end hosts not supporting E2ENAT within a private network, an E2ENAT gateway may behave also as an legacy NAT gateway. Packets from such hosts can be distinguished by the gateway as the packets have private source addresses and public destination addresses, which is not the case of packets from E2ENAT aware hosts.

A NAT gateway may also support legacy UPnP.

So, introduction of E2ENAT is no worse than introduction of legacy NAT.

2.6 DNS with End to End NAT

Domain name for a shared public address may be looked up as usual through DNS:

```
www.example.com A 208.77.188.166
166.188.77.208.in-addr.arpa PTR www.example.com
```

Moreover, each end host sharing the public address may individually have its own CNAME identified by port numbers:

```
p1.example.org CNAME www.example.com
1.0.166.188.77.208.in-addr.arpa PTR p1.example.org
p2.example.org CNAME www.example.com
2.0.166.188.77.208.in-addr.arpa PTR p2.example.org
```

Though RFC1034 gives an example with PTR that indirection should point to canonical names [RFC1034], the reason is to prevent

automatic indirection, which is not the case with PTR. As the RFC encourages robustness for chained indirection, there should be no practical problem.

DNS query from end hosts should be relayed by NAT gateways with source port number randomization to increase entropy against birthday attacks.

2.7 Multicast with End to End NAT

As we are not running out of multicast addresses, multicast addresses should be shared unmodified between a public and a private networks.

As multicast routing, in general, is affected by reverse path to a source address, which is, in a private network, directed to a NAT gateway, end hosts should not directly send multicast packets but unicast them IP over IP (using, for example, register message of PIM-SM [PIM]) to the NAT gateway to ask the gateway relaying. Protocol other than PIM-SM may be used within the private network. An end host can not be a rendez-vous point of PIM-SM nor a core of CBT.

2.8 Mobility with End to End NAT

Mobile IP [MIP] needs some modification to accommodate E2ENAT. A mobile host must be able to know NAT configuration of its home network.

Moreover, if a mobile host is in a public network of a NAT gateway, it can not, in general, use port numbers assigned at the home network. So, mobile IP must be extended to use IP over UDP over IP for tunneling and to register to a home agent not only foreign addresses but also foreign port numbers. Then, a mobile host needs only a single port number in foreign environment for its operation.

3. Port Number Considerations

Though E2ENAT is almost fully transparent to upper layers, it is still NAT.

So, it is required that packets to end hosts sharing an IP address must, somehow, be demultiplexed by a NAT gateway. However, as the demultiplexing information is not located in payload part of IP packets, it depends on a protocol above IP.

It should be noted that a NAT gateway and all the end hosts behind it must agree on how the demultiplexing information is extracted that strong standardization is necessary here. Thus, it must be assumed that the protocol is identified by IANA assigned value in protocol

field of IP headers, ignoring a possibility of private agreement on the value.

For most, if not all, true transport protocols (such as TCP, UDP, DCCP, SCTP, UDPLite), demultiplexing depends on destination port numbers located at the third and the forth bytes of transport headers.

The same location may be usable for some protocols. For example, ESP packets may be demultiplexed based on the lower 16 bit of SPI. To do so, a destination host must be able to control the lower 16 bit of SPI.

Other protocols may use different location, for which E2ENAT may provide special care. Considering that an ICMP packet generated against an IP packet contains only the first 64 bits of payload of the original packet, demultiplexing information for source transport is implicitly assumed to be located in the 64 bits. So, it is natural that demultiplexing information for destination transport is also located in the same field. If the information is 16 bit word and 16 bit aligned, there are only four variations on the location. This memo assumes so.

For example, AH packets may be demultiplexed as if the lower 16 bit of SPI, located at the seventh and eighth bytes of a payload, is a destination port number.

To demultiplex ICMP packets containing original packets causing the ICMP packets, source port numbers (including upper 16 bit of SPI) of the original packets should be used instead of destination ones.

Identifier and sequence number fields of ICMP Echo, Timestamp and Information Request may be used for demultiplexing as if they are source and destination port numbers.

In this memo, the demultiplexing information is assumed to be 16 bit long and is called port number.

4. Implementation Status

Gateways and end hosts for E2ENAT are implemented on NETBSD 5.0 and are working.

Though code for address translation is short and simple, restricting source port number needs several hundred lines of coding.

It is confirmed that ftp port command works on ftp clients behind a NAT gateway.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

Though section 2.2 requires an end host use source port numbers assigned to it, ignoring the restriction makes the host can not receive reply and is only as harmful as a host with forged source addresses. Three way handshaking is applicable here.

DNS reverse lookup discussed in section 2.6 enables access control by source addresses and port numbers represented in domain names. Source port randomization in section 2.6 protects against birthday attacks.

As is discussed in section 3, IPSEC works over E2ENAT gateways, as long as SPI is properly constrained.

7. Informative References

[IP] J. Postel (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, September 1981.

[SALTZER] J.H. Saltzer, D.P.Reed, D.D.Clark, "End-To-End Arguments in System Design", ACM TOCS, V. 2, N. 4, pp. 277-288, November 1984.

[ICMP] J. Postel, "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", RFC 791, September 1981.

[RFC1034] P. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, November 1987.

[PIM] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, August 2006.

[MIP] C. Perkins (ed.), "IP Mobility Support", RFC 2001, October 1996.

Author's Address

Masataka Ohta
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
2-12-1, O-okayama, Meguro-ku
Tokyo 152-8552, JAPAN

Phone: +81-3-5734-3299
Fax: +81-3-5734-3299
EMail: mohta@necom830.hpcl.titech.ac.jp