

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 5, 2012

M. Boucadair
France Telecom
J. Touch
USC/ISI
P. Levis
France Telecom
R. Penno
Juniper Networks
September 2, 2011

Analysis of Solution Candidates to Reveal a Host Identifier in Shared
Address Deployments
draft-boucadair-intarea-nat-reveal-analysis-04

Abstract

This document analyzes a set of solution candidates which have been proposed to mitigate some of the issues encountered when address sharing is used. In particular, this document focuses on means to reveal a host identifier when a Carrier Grade NAT (CGN) or application proxies are involved in the path. This host identifier must be unique to each host under the same shared IP address.

The ultimate goal is to assess the viability of proposed solutions and hopefully to make a recommendation on the more suitable solution(s).

Requirements Language

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].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on March 5, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
1.1. Problem to Be Solved	4
1.2. HOST_ID and Privacy	5
1.3. IPv6 May Also Be Concerned	6
1.4. Purpose and Scope	6
2. Recommendations	6
3. Solutions Analysis	8
3.1. Define an IP Option	8
3.1.1. Description	8
3.1.2. Analysis	9
3.2. Define a TCP Option	9
3.2.1. Description	9
3.2.2. Analysis	9
3.3. Use the Identification Field of IP Header (IP-ID)	10
3.3.1. Description	10
3.3.2. Analysis	11
3.4. Inject Application Headers	11
3.4.1. Description	11
3.4.2. Analysis	11
3.5. PROXY Protocol	12
3.5.1. Description	12
3.5.2. Analysis	12
3.6. Enforce a Source-based Selection Algorithm at the Server Side (Port Set)	12
3.6.1. Description	12
3.6.2. Analysis	13
3.7. Host Identity Protocol (HIP)	13
3.7.1. Description	13
3.7.2. Analysis	13
4. IANA Considerations	13
5. Security Considerations	14
6. Acknowledgments	14
7. References	14
7.1. Normative References	14
7.2. Informative References	14
Authors' Addresses	16

1. Introduction

As reported in [RFC6269], several issues are encountered when an IP address is shared among several subscribers. Examples of such issues are listed below:

- o Implicit identification (Section 13.2 of [RFC6269])
- o SPAM (Section 13.3 of [RFC6269])
- o Blacklisting a mis-behaving user (Section 13.1 of [RFC6269])
- o Redirect users with infected machines to a dedicated portal (Section 5.1 of [RFC6269])

The sole use of the IPv4 address is not sufficient to uniquely distinguish a host. As a mitigation, it is tempting to investigate means which would help in disclosing an information to be used by the remote server as a means to uniquely disambiguate packets of hosts using the same IPv4 address.

The risk of not mitigating these issues are: OPEX increase for IP connectivity service providers (costs induced by calls to a hotline), revenue loss for content providers (loss of users audience), customers unsatisfaction (low quality of experience, service segregation, etc.).

1.1. Problem to Be Solved

Observation: Today, servers use the source IPv4 address as an identifier to treat some incoming connections differently. Tomorrow, due to the introduction of CGNs (e.g., NAT44 [I-D.ietf-behave-lsn-requirements], NAT64 [RFC6146]), that address will be shared. In particular, when a server receives packets from the same source address. Because this address is shared, the server does not know which host is the sending host.

Objective: The server should be able to sort out the packets by sending host.

Requirement: The server must have extra information than the source IP address to differentiate the sending host. We call HOST_ID this information.

For all solutions analyzed, we provide answers to the following questions:

What is the HOST_ID? It must be unique to each host under the same IP address. It does not need to be globally unique. Of course, the combination of the (public) IPv4 source address and the identifier (i.e., HOST_ID) ends up being relatively unique. As unique as today's 32-bit IPv4 addresses which, today, can change

when a host re-connects.

Where is the HOST_ID? (which protocol, which field): If the HOST_ID is put at the IP level, all packets will have to bear the identifier. If it is put at a higher connection-oriented level, the identifier is only needed once in the session establishment phase (for instance TCP three-way-handshake), then, all packets received in this session will be attributed to the HOST_ID designated during the session opening.

Who puts the HOST_ID? For almost all the analyzed solutions, the address sharing function injects the HOST_ID. When there are several address sharing functions in the data path, we describe to what extent the proposed solution is efficient. Another option to avoid potential performance degradation is to let the host inject its HOST_ID but the address sharing function will check its content (just like an IP anti-spoofing function).

What are the security considerations? Security considerations are common to all analyzed solutions (see Section 5). Privacy-related aspect are discussed in Section 1.2.

1.2. HOST_ID and Privacy

HOST_ID provides an additional information to uniquely disambiguate a host among those sharing the same IP address. Unlike URIs, HOST_ID does not leak user's identity information.

The HOST_ID does not reveal more privacy information than what the source IP address does in a non-shared address environment (see [I-D.morris-privacy-considerations]).

The volatility of the HOST_ID information is similar to the source IP address: a distinct HOST_ID may be used by the address sharing function when the host reboots or gets a new internal IP address. If the HOST_ID is persistent it may be used to track a host (similar to persistent IP addresses).

The trust on the information conveyed in the HOST_ID is likely to be the same as for current practices with the source IP address. In that sense, a HOST_ID can be spoofed as this is also the case for spoofing an IP address.

It is the responsibility of the remote server to rely or not on the content of the HOST_ID to enforce its policies and to log or not the content conveyed in the HOST_ID.

Enabling explicit identification means an adequate security suite is

more robust than relying on source IP address or HOST_ID. But tension may appear between strong privacy and usability (see Section 4.2 of [I-D.iab-privacy-workshop]).

1.3. IPv6 May Also Be Concerned

Issues similar to the ones described in Section 1.1 may be encountered also in an IPv6 environment (e.g., when the same /64 is used among several hosts).

1.4. Purpose and Scope

The purpose of this document is to analyze the solutions that have been proposed so far and to assess to what extent they solve the problem (see Section 1.1).

The purpose of this document is not to argue in favor of mandating the use of a HOST_ID but to document encountered issues, proposed solutions and their limitations.

Only IPv4-based solutions are analyzed in the following sections:

- o define a new IP option (Section 3.1)
- o define a new TCP option (Section 3.2)
- o use the Identification field of IP header (denoted as IP-ID, Section 3.3)
- o inject application headers (Section 3.4)
- o enable Proxy Protocol ((Section 3.5))
- o use of port set (Section 3.6)
- o activate HIP (Section 3.7).

2. Recommendations

The following Table 1 summarizes the approaches analyzed in this document.

- o "Success ratio" indicates the ratio of successful communications when the option is used. Provided figures are inspired from the results documented in [Options].
- o "Deployable today" indicates if the solution can be generalized without any constraint on current architectures and practices.
- o "Possible Perf Impact" indicates the level of expected performance degradation. The rationale behind the indicated potential performance degradation is whether the injection requires some treatment at the IP level or not.

- o "OS TCP/IP Modif" indicates whether a modification of the OS TCP/IP stack is required at the server side.

	IP Option	TCP Option	IP-ID	HTTP Header (XFF)	Proxy Protocol	Port Set	HIP
UDP	Yes	No	Yes	No	No	Yes	
TCP	Yes	Yes	Yes	No	Yes	Yes	
HTTP	Yes	Yes	Yes	Yes	Yes	Yes	
Encrypted Traffic	Yes	Yes	Yes	No	Yes	Yes	
Success Ratio	30%	99%	100%	100%	Low	100%	Low
Possible Perf Impact	High	Med to High	Low to Med	Med to High	High	No	N/A
OS TCP/IP Modif	Yes	Yes	Yes	No	No	No	
Deployable Today	Yes	Yes	Yes	Yes	No	Yes	No
Notes			(1)	(2)		(1) (3)	(4) (5)

Table 1: Summary of analyzed solutions.

Notes for the above table:

- (1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record)
- (2) This solution is widely deployed
- (3) When the port set is not advertised, the solution is less efficient for third-party services.
- (4) Requires the client and the server to be HIP-compliant and HIP infrastructure to be deployed.

- (5) If the client and the server are HIP-enabled, the address sharing function does not need to insert a host-hint. If the client is not HIP-enabled, designing the device that performs address sharing to act as a UDP/TCP-HIP relay is not viable.

According to the above table and the analysis elaborated in Section 3:

- o IP Option, IP-ID and Proxy Protocol proposals are broken;
- o HIP is not largely deployed;
- o The use of Port Set may contradict the port randomization [RFC6056] requirement identified in [RFC6269]. This solution can be used by a service provider for the delivery of its own service offerings relying on implicit identification.
- o XFF is de facto standard deployed and supported in operational networks (e.g., HTTP Servers, Load-Balancers, etc.).
- o From an application standpoint, the TCP Option is superior to XFF since it is not restricted to HTTP. Nevertheless XFF is compatible with the presence of address sharing and load-balancers in the communication path. To provide a similar functionality, the TCP Option may be extended to allow conveying a list of IP addresses to not loose the source IP address in the presence of load-balancers. Note that TCP Option requires the modification of the OS TCP/IP stack of remote servers; which can be seen as a blocking point.

As a conclusion of this analysis, the following recommendation is made:

[Hopefully to be completed]

3. Solutions Analysis

3.1. Define an IP Option

3.1.1. Description

This proposal aims to define an IP option [RFC0791] to convey a "host identifier". This identifier can be inserted by the address sharing function to uniquely distinguish a host among those sharing the same IP address. The option can convey an IPv4 address, the prefix part of an IPv6 address, etc.

Another way for using IP option has been described in Section 4.6 of [RFC3022].

3.1.2. Analysis

Unlike the solution presented in Section 3.2, this proposal can apply for any transport protocol. Nevertheless, it is widely known that routers (and other middle boxes) filter IP options. IP packets with IP options can be dropped by some IP nodes. Previous studies demonstrated that "IP Options are not an option" (Refer to [Not_An_Option], [Options]).

As a conclusion, using an IP option to convey a host-hint is not viable.

3.2. Define a TCP Option

3.2.1. Description

This proposal [I-D.wing-nat-reveal-option] defines a new TCP option called USER_HINT. This option encloses the TCP client's identifier (e.g., the lower 16 bits of their IPv4 address, their VLAN ID, VRF ID, subscriber ID). The address sharing device inserts this TCP option to the TCP SYN packet.

3.2.2. Analysis

The risk related to handling a new TCP option is low as measured in [Options].

[I-D.wing-nat-reveal-option] discusses the interference with other TCP options.

Using a new TCP option to convey the host-hint does not require any modification to the applications but it is applicable only for TCP-based applications. Applications relying on other transport protocols are therefore left unsolved.

Some downsides have been raised against defining a TCP option to reveal a host identity:

- o Conveying an IP address in a TCP option may be seen as a violation of OSI layers but since IP addresses are already used for the checksum computation, this is not seen as a blocking point. Moreover, Updated version of [I-D.wing-nat-reveal-option] does not allow anymore to convey an IP address (the HOST_ID is encoded in 16bits).

- o TCP option space is limited, and might be consumed by the TCP client. Earlier versions of [I-D.wing-nat-reveal-option] discuss two approaches to sending the HOST_ID: sending the HOST_ID in the TCP SYN (which consumes more bytes in the TCP header of the TCP SYN) and sending the HOST_ID in a TCP ACK (which consumes only two bytes in the TCP SYN). Content providers may find it more desirable to receive the HOST_ID in the TCP SYN, as that more closely preserves the host hint received in the source IP address as per current practices. It is more complicated to implement sending the HOST_ID in a TCP ACK, as it can introduce MTU issues if the ACK packet also contains TCP data, or a TCP segment is lost. The latest specification of the HOST_ID TCP Option, documented at [I-D.wing-nat-reveal-option], allows only to enclose the HOST_ID in the TCP SYN packet.
- o When there are several NATs in the path, the original HOST_ID may be lost. In such case, the procedure may not be efficient.
- o Interference with current usages such as X-Forwarded-For (see Section 3.4) should be elaborated to specify the behavior of servers when both options are used; in particular specify which information to use: the content of the TCP option or what is conveyed in the application headers.
- o When load-balancers or proxies are in the path, this option does not allow to preserve the original source IP address and source. Preserving such information is required for logging purposes for instance.

3.3. Use the Identification Field of IP Header (IP-ID)

3.3.1. Description

IP-ID (Identification field of IP header) can be used to insert an information which uniquely distinguishes a host among those sharing the same IPv4 address. An address sharing function can re-write the IP-ID field to insert a value unique to the host (16 bits are sufficient to uniquely disambiguate hosts sharing the same IP address). Note that this field is not altered by some NATs; hence some side effects such as counting hosts behind a NAT as reported in [Count].

A variant of this approach relies upon the format of certain packets, such as TCP SYN, where the IP-ID can be modified to contain a 16 bit host-hint. Address sharing devices performing this function would require to indicate they are performing this function out of band, possibly using a special DNS record.

3.3.2. Analysis

This usage is not compliant with what is recommended in [I-D.ietf-intarea-ipv4-id-update].

3.4. Inject Application Headers

3.4.1. Description

Another option is to not require any change at the transport nor the IP levels but to convey at the application payload the required information which will be used to disambiguate hosts. This format and the related semantics depend on its application (e.g., HTTP, SIP, SMTP, etc.).

For HTTP, the X-Forwarded-For (XFF) header can be used to display the original IP address when an address sharing device is involved. Service Providers operating address sharing devices can enable the feature of injecting the XFF header which will enclose the original IPv4 address or the IPv6 prefix part. The address sharing device has to strip all included XFF headers before injecting their own. Servers may rely on the contents of this field to enforce some policies such as blacklisting misbehaving users. Note that XFF can also be logged by some servers (this is for instance supported by Apache).

3.4.2. Analysis

Not all applications impacted by the address sharing can support the ability to disclose the original IP address. Only a subset of protocols (e.g., HTTP) can rely on this solution.

For the HTTP case, to prevent users injecting invalid host-hints, an initiative has been launched to maintain a list of trusted ISPs using XFF: See for example the list available at: [Trusted_ISPs] of trusted ISPs as maintained by Wikipedia. If an address sharing device is on the trusted XFF ISPs list, users editing Wikipedia located behind the address sharing device will appear to be editing from their "original" IP address and not from the NATed IP address. If an offending activity is detected, individual hosts can be blacklisted instead of all hosts sharing the same IP address.

XFF header injection is a common practice of load balancers. When a load balancer is in the path, the original content of any included XFF header should not be stripped. Otherwise the information about the "origin" IP address will be lost.

When several address sharing devices are crossed, XFF header can

convey the list of IP addresses. The origin HOST_ID can be exposed to the target server.

XFF also introduces some implementation complexity if the HTTP packet is at or close to the MTU size.

It has been reported that some "poor" implementation may encounter some parsing issues when injecting XFF header.

For encrypted HTTP traffic, injecting XFF header may be broken.

3.5. PROXY Protocol

3.5.1. Description

The solution, referred to as Proxy Protocol [Proxy], does not require any application-specific knowledge. The rationale behind this solution is to prepend each connection with a line reporting the characteristics of the other side's connection as shown in the example below (excerpt from [Proxy]):

```
PROXY TCP4 198.51.100.1 198.51.100.11 56324 443\r\n
```

Upon receipt of a message conveying this line, the server removes the line. The line is parsed to retrieve the transported protocol. The content of this line is recorded in logs and used to enforce policies.

3.5.2. Analysis

This solution can be deployed in a controlled environment but it can not be deployed to all access services available in the Internet. If the remote server does not support the Proxy Protocol, the session will fail. Other complications will raise due to the presence of firewalls for instance.

As a consequence, this solution is broken and can not be recommended.

3.6. Enforce a Source-based Selection Algorithm at the Server Side (Port Set)

3.6.1. Description

This solution proposal does not require any action from the address sharing function to disclose a host identifier. Instead of assuming all the ports are associated with the same host, a random-based algorithm (or any port selection method) is run to generate the set of ports (including the source port of the received packet). The

length of the ports set to be generated by the server may be configurable (e.g., 8, 32, 64, 512, 1024, etc.). Instead of a random-based scheme, the server can use contiguous port ranges to form the port sets.

The server may reduce (or enlarge) the width of the ports set of the misbehaving action is (not) mitigated.

A variant of this proposal is to announce by off-line means the port set assignment policy of an operator. This announcement is not required for the delivery of internal services (i.e., offered by the service provider deploying the address sharing function) relying on implicit identification.

3.6.2. Analysis

In nominal mode, no coordination is required between the address sharing function and the server side but the efficiency of the method depends on the port set selection algorithm.

The method is more efficient if the provider that operates the address sharing device advertises its port assignment policy but this may contradicts the port randomization as identified in [RFC6269].

The method is deterministic for the delivery of services offered by the service provider offering also the IP connectivity service.

3.7. Host Identity Protocol (HIP)

3.7.1. Description

[RFC5201] specifies an architecture which introduces a new namespace to convey an identity information.

3.7.2. Analysis

This solution requires both the client and the server to support HIP [RFC5201]. Additional architectural considerations are to be taken into account such as the key exchanges, etc.

If the address sharing function is required to act as a UDP/TCP-HIP relay, this is not a viable option.

4. IANA Considerations

This document does not require any action from IANA.

5. Security Considerations

The same security concerns apply for the injection of an IP option, TCP option and application-related content (e.g., XFF) by the address sharing device. If the server trusts the content of the HOST_ID field, a third party user can be impacted by a misbehaving user to reveal a "faked" original IP address.

6. Acknowledgments

Many thanks to D. Wing and C. Jacquenet for their review, comments and inputs.

Thanks also to P. McCann, T. Tsou, Z. Dong, B. Briscoe, T. Taylor, M. Blanchet, D. Wing and A. Yourtchenko for the discussions in Prague.

Some of the issues related to defining a new TCP option have been raised by L. Eggert.

7. References

7.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

7.2. Informative References

- [Count] "A technique for counting NATted hosts",
<<http://www.cs.columbia.edu/~smb/papers/fnat.pdf>>.
- [I-D.iab-privacy-workshop] Cooper, A., "Report from the Internet Privacy Workshop",
draft-iab-privacy-workshop-00 (work in progress),
June 2011.

- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "Common requirements for Carrier Grade NAT
(CGN)", draft-ietf-behave-lsn-requirements-03 (work in
progress), August 2011.
- [I-D.ietf-intarea-ipv4-id-update]
Touch, J., "Updated Specification of the IPv4 ID Field",
draft-ietf-intarea-ipv4-id-update-02 (work in progress),
March 2011.
- [I-D.morris-privacy-considerations]
Aboba, B., Morris, J., Peterson, J., and H. Tschofenig,
"Privacy Considerations for Internet Protocols",
draft-morris-privacy-considerations-03 (work in progress),
March 2011.
- [I-D.wing-nat-reveal-option]
Yourtchenko, A. and D. Wing, "Revealing hosts sharing an
IP address using TCP option",
draft-wing-nat-reveal-option-02 (work in progress),
June 2011.
- [Not_An_Option]
R. Fonseca, G. Porter, R. Katz, S. Shenker, and I.
Stoica,, "IP options are not an option", 2005, <[http://
www.eecs.berkeley.edu/Pubs/TechRpts/2005/
EECS-2005-24.html](http://www.eecs.berkeley.edu/Pubs/TechRpts/2005/EECS-2005-24.html)>.
- [Options] Alberto Medina, Mark Allman, Sally Floyd, "Measuring
Interactions Between Transport Protocols and Middleboxes",
2005, <[http://conferences.sigcomm.org/imc/2004/papers/
p336-medina.pdf](http://conferences.sigcomm.org/imc/2004/papers/p336-medina.pdf)>.
- [Proxy] Tarreau, W., "The PROXY protocol", November 2010, <[http://
haproxy.lwt.eu/download/1.5/doc/proxy-protocol.txt](http://haproxy.lwt.eu/download/1.5/doc/proxy-protocol.txt)>.
- [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson,
"Host Identity Protocol", RFC 5201, April 2008.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful
NAT64: Network Address and Protocol Translation from IPv6
Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P.
Roberts, "Issues with IP Address Sharing", RFC 6269,
June 2011.

[Trusted_ISPs]

"Trusted XFF list", <http://meta.wikimedia.org/wiki/XFF_project#Trusted_XFF_list>.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Joe Touch
USC/ISI

Email: touch@isi.edu

Pierre Levis
France Telecom
Caen, 14000
France

Email: pierre.levis@orange-ftgroup.com

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net

