INTAREA WG M. Boucadair
Internet-Draft France Telecom
Intended status: Informational J. Touch
Expires: September 12, 2013 USC/ISI
P. Levis
France Telecom
R. Penno
Cisco
March 11, 2013

Analysis of Solution Candidates to Reveal a Host Identifier (HOST_ID) in Shared Address Deployments

draft-ietf-intarea-nat-reveal-analysis-06

Abstract

This document is a collection of solutions to reveal a host identifier (denoted as HOST_ID) when a Carrier Grade NAT (CGN) or application proxies are involved in the path. This host identifier is used by a remote server to sort out the packets by sending host. The host identifier must be unique to each host under the same shared IP address.

This document analyzes a set of solution candidates to reveal a host identifier; no recommendation is sketched in the document.

Table of Contents

1. Introduction

As reported in [RFC6269], several issues are encountered when an IP address is shared among several subscribers. These issues are encountered in various deployment contexts: e.g., Carrier Grade NAT (CGN), application proxies or A+P [RFC6346]. Examples of such issues are: implicit identification (Section 13.2 of [RFC6269]), SPAM (Section 13.3 of [RFC6269]), blacklisting a mis-behaving host (Section 13.1 of [RFC6269]) or redirect users with infected machines to a dedicated portal (Section 5.1 of [RFC6269]).

In particular, some servers use the source IPv4 address as an identifier to treat some incoming connections differently. Due to the deployment of CGNs (e.g., NAT44 [RFC3022], 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 [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 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 include: OPEX (Operational Expenditure) increase for IP connectivity service providers (costs induced by calls to a hotline), revenue loss for content providers (loss of users audience) and customers' dissatisfaction (low quality of experience, service segregation, etc.).

The purpose of this document is to analyze a set of alternative channels to convey a host identifier and to assess to what extent they solve the problem described in Section 2. Below are listed the alternatives analyzed in the document:

- o Use the Identification field of IP header (denoted as IP-ID, Section 4.1).
- o Define a new IP option (Section 4.2).
- o Define a new TCP Option (Section 4.3).
- o Inject application headers (Section 4.4).
- o Enable Proxy Protocol (Section 4.5).
- o Assign port sets (Section 4.6).
- o Activate HIP (Host Identity Protocol, Section 4.7).
- o Use a notification channel (Section 4.8).

o Use an out-of-band mechanism (Section 4.9).

A synthesis is provided in Section 5 while the detailed analysis is elaborated in Section 4.

Section 3 discusses privacy issues common to all candidate solutions. It is out of scope of this document to elaborate on privacy issues specific to each solution.

2. On HOST_ID

Policies relying on source IP address which are enforced by some servers will be applied to all hosts sharing the same IP address. For example, blacklisting the IP address of a spammer host will result in all other hosts sharing that address having their access to the requested service restricted. [RFC6269] describes the issues in detail. Therefore, due to address sharing, servers need extra information beyond the source IP address to differentiate the sending host. We call this information the HOST_ID.

HOST_ID is not designed to reveal the identity of a user, a subscriber, or an application. HOST_ID is designed to identify a host under a shared IP address.

Because HOST_ID is used by a remote server to sort out the packets by sending host, HOST_ID must be unique to each host under the same IP address. HOST_ID does not need to be globally unique. Of course, the combination of the (public) IP source address and the identifier (i.e., HOST_ID) ends up being unique.

If the HOST_ID is conveyed at the IP level, all packets will have to bear the identifier. If it is conveyed 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.

Within this document, we assume the address-sharing function injects the HOST_ID. Another deployment 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). For some proposals, the HOST_ID is retrieved using an out-of-band mechanism or signaled in a dedicated notification channel.

Security considerations are common to all analyzed solutions (see Section 7). Privacy-related aspects are discussed in Section 3.

3. HOST_ID and Privacy

IP address sharing is motivated by a number of different factors. For years, many network operators have conserved the use of public IPv4 addresses by making use of Customer Premises Equipment (CPE) that assigns a single public IPv4 address to all hosts within the customer's local area network and uses NAT [RFC3022] to translate between locally unique private IPv4 addresses and the CPE's public address. With the exhaustion of IPv4 address space, address sharing between customers on a much larger scale is likely to become much more prevalent. While many individual users are unaware of and uninvolved in decisions about whether their unique IPv4 addresses get revealed when they send data via IP, some users realize privacy benefits associated with IP address sharing, and some may even take steps to ensure that NAT functionality sits between them and the public Internet. IP address sharing makes the actions of all users behind the NAT function unattributable to any single host, creating room for abuse but also providing some identity protection for non-abusive users who wish to transmit data with reduced risk of being uniquely identified.

The proposals considered in this document add a measure of identifiability back to hosts that share a public IP address. The extent of that uniqueness depends on what information is included in the HOST_ID.

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. As with persistent IP addresses, persistent HOST_IDs facilitate user tracking over time.

As a general matter, the HOST_ID proposals do not seek to make hosts any more identifiable than they would be if they were using a public, non-shared IP address. However, depending on the solution proposal, the addition of HOST_ID information may allow a device to be fingerprinted more easily than it otherwise would be. Should multiple solutions be combined (e.g., TCP Option and Forwarded header) that include different pieces of information in the HOST_ID, fingerprinting may become even easier.

A HOST_ID can be spoofed as this is also the case for spoofing an IP address. Furthermore, users of network-based anonymity services (like Tor) may be capable of stripping HOST_ID information before it reaches its destination.

HOST_ID specification document(s) should explain the privacy impact of the solutions they specify, including the extent of HOST_ID

uniqueness and persistence, assumptions made about the lifetime of the HOST_ID, whether and how the HOST_ID can be obfuscated or recycled, and the impact of the use of the HOST_ID on device or implementation fingerprinting. [I-D.iab-privacy-considerations] provides further guidance.

For more discussion about privacy, refer to [RFC6462].

3.1. Privacy-related Considerations

Whatever the channel used to convey the HOST_ID, the following design consideration are to be taken into account:

Uniqueness of identifiers in HOST_ID: It is recommended that HOST_IDs be limited to providing local uniqueness rather than global uniqueness.

Refresh rate of HOST_ID: Address-sharing function should not use permanent HOST_ID values.

Manipulate HOST_IDs: Address-sharing function should be able to strip, rewrite and add HOST_ID fields.

Interference between HOST_IDs: An address-sharing function, able to inject HOST_IDs in several layers, should reveal subsets of the same information (e.g., full IP address, lower 16 bits of IP address, etc.).

4. Detailed Solutions Analysis

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

4.1.1. Description

IP-ID (Identification field of IP header) can be used to insert 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, there are some side effects such as counting hosts behind a NAT as reported in [Count].

The address-sharing function injecting the HOST_ID must follow the rules defined in [RFC6864]; in particular the same HOST_ID is not re-assigned to another host sharing the same IP address during a given time interval.

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_ID.

Address-sharing devices using this solution would be required to indicate that out of band, possibly using a special DNS record.

4.1.2. Analysis

This usage is not consistent with the fragment reassembly use of the Identification field [RFC0791] or the updated handling rules for the Identification field [RFC6864].

Complications may arise if the packet is fragmented before reaching the device injecting the HOST_ID. To appropriately handle those packets, the address-sharing function will need to maintain a lot of state.

Another complication to be encountered is where translation is balanced among several NATs; setting the appropriate HOST_ID by a given NAT would alter the coordination between those NATs. Of course, one can argue this coordinated NAT scenario is not a typical deployment scenario but still using IP-ID as a channel to convey a HOST_ID is ill-advised.

4.2. Define an IP Option

4.2.1. Description

A solution alternative to convey the HOST_ID is to define an IP option [RFC0791]. A HOST_ID IP option can be inserted by the address-sharing function to uniquely distinguish a host among those sharing the same IP address. An example of such option is documented in [I-D.chen-intarea-v4-uid-header-option]. This IP option allows the conveyance of an IPv4 address, an IPv6 prefix, a GRE (Generic Routing Encapsulation) key, an IPv6 Flow Label, etc.

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

4.2.2. Analysis

This proposal can apply to any transport protocol. Nevertheless, it is widely known that routers and other middleboxes filter IP options (e.g., drop IP packets with unknown IP options, strip unknown IP options, etc.). Previous studies demonstrated that "IP Options are not an option" (Refer to [Not_An_Option], [Options]).

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

4.3. Define a TCP Option

4.3.1. Description

HOST_ID may be conveyed in a dedicated TCP Option. An example is specified in [I-D.wing-nat-reveal-option]. This option encloses the TCP client's identifier (e.g., the lower 16 bits of its IPv4 address, its VLAN ID, VRF ID, or subscriber ID). The address-sharing device inserts this TCP Option into the TCP SYN packet.

4.3.2. Analysis

Using a new TCP Option to convey the HOST_ID 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.

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

The risk to experience session failures due to handling a new TCP Option is low as measured in [Options]. [I-D.abdo-hostid-tcpopt-implementation] provides a detailed implementation and experimentation report of a HOST_ID TCP Option. [I-D.abdo-hostid-tcpopt-implementation] investigated in depth the impact of activation HOST_ID on the host, the address-sharing function, and the enforcement of policies at the server side. [I-D.abdo-hostid-tcpopt-implementation] reports a failure ratio of 0.103% among top 100000 websites.

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] no longer allows conveyance of a full IP address as the HOST_ID is encoded in 16 bits.

- o TCP Option space is limited and might be consumed by the TCP client. [I-D.abdo-hostid-tcpopt-implementation] discusses 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_ID 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. Note [I-D.wing-nat-reveal-option] only allows enclosing the HOST_ID in the TCP SYN packet.

- When there are several NATs in the path, the original HOST_ID may be lost. The loss of the original HOST_ID may not be a problem as the target usage is between proxies or a CGN and server. Only the information leaked in the last communication leg (i.e., between the last address-sharing function and the server) is likely to be useful.

- Interference with usages such as Forwarded HTTP header (see Section 4.4) 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.

- When load-balancers or proxies are in the path, this option does not allow the preservation of the original source IP address and source port. Preserving such information is required for logging purposes for instance (e.g., [RFC6302]). [I-D.abdo-hostid-tcpopt-implementation] defines a TCP Option which allows revealing various combinations of source information (e.g., source port, source port and source IP address, source IPv6 prefix, etc.).

More discussion about issues raised when extending TCP can be found at [ExtendTCP].

4.4. Inject Application Protocol Message Headers

4.4.1. Description

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

For HTTP, Forwarded header ([I-D.ietf-appsawg-http-forwarded]) 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 Forwarded header

which will enclose the original IPv4 address or the IPv6 prefix part (see the example shown in Figure 1). The address-sharing device has to strip all included Forwarded headers before injecting its own. Servers may rely on the contents of this field to enforce some policies such as blacklisting misbehaving users.

Note that the X-Forwarded-For (XFF) header is obsoleted by [I-D.ietf-appsawg-http-forwarded].

```
Forwarded: for=192.0.2.1,for=[2001:db8::1]
Forwarded: proto=https;by=192.0.2.15
```

Figure 1: Example of Forwarded-For

4.4.2. Analysis

Not all applications impacted by 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_IDs, an initiative has been launched by Wikipedia to maintain a list of trusted ISPs (Internet Service Providers) using XFF (See the list available at [Trusted_ISPs]). 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, the Forwarded header can convey the list of IP addresses (e.g., Figure 1). The origin HOST_ID can be exposed to the target server.

Injecting Forwarded header also introduces some implementations complexity if the HTTP message is at or close to the MTU size.

It has been reported that "poor" HTTP proxy implementations may encounter parsing issues when injecting an XFF header.

Injecting Forwarded header for all encrypted HTTP traffic is infeasible.

4.5. PROXY Protocol

4.5.1. 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 depicted in Figure 2. The header line shown in this example is for a TCP over IPv4 connection received from 192.0.2.1:56324 and destined to 192.0.2.15:443. The "PROXY" string is used to identify the Proxy Protocol while "\r\n" indicates CRLF.

```
PROXY TCP4 192.0.2.1 192.0.2.15 56324 443\r\n
```

Figure 2: Example of PROXY conection report

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.

4.5.2. 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 arise due to the presence of firewalls, for instance.

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

4.6. Assign Port Sets

4.6.1. Description

This solution does not require any action from the address-sharing function to disclose a host identifier. Instead of assuming all transport ports are associated with one single host, each host under the same external IP address is assigned a restricted port set. These port sets are then advertised to remote servers using off-line means. 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.

Port sets assigned to hosts may be static or dynamic.

Port set announcements to remote servers are not required to reveal the identity of individual hosts but only to advertise the enforced policy to generate non-overlapping port sets (e.g., the transport space associated with an IP address is fragmented to contiguous blocks of 2048 port numbers).

An example of such an option is documented in [RFC6346].

4.6.2. Analysis

The solution does not require defining new fields nor options; it is policy-based.

The solution may contradict the port randomization ([RFC6056]) as identified in [RFC6269]. A mitigation would be to avoid assigning static port sets to individual hosts.

The method is convenient for the delivery of services offered by the service provider also offering the Internet access service.

4.7. Host Identity Protocol (HIP)

4.7.1. Description

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

4.7.2. 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.

An alternative deployment model, which does not require the client to be HIP-enabled, is having the address-sharing function behave as a UDP/TCP-HIP relay. This model is also not viable as it assumes all servers are HIP-enabled.

4.8. Use of a Notification Channel (e.g., ICMP)

4.8.1. Description

Another alternative is to convey the HOST_ID using a separate notification channel than the packets issued to invoke the service.

An implementation example is defined in [I-D.yourtchenko-nat-reveal-ping]. This solution relies on a mechanism where the address-sharing function encapsulates the

necessary host-identifying information into an ICMP Echo Request packet that it sends in parallel with the initial session creation (e.g., SYN). The information included in the ICMP Request Data portion describes the five-tuples as seen on both of the sides of the address-sharing function.

4.8.2. Analysis

- o This ICMP proposal is valid for any transport protocol that uses a port number. The address-sharing function may be configured with the transport protocols which will trigger issuing those ICMP messages.
- o A hint should be provided to the ultimate server (or intermediate nodes) that the ICMP Echo Request conveys a HOST_ID. This may be implemented using magic numbers.
- o Even if ICMP packets are blocked in the communication path, the user connection does not have to be impacted.
- o Implementations requiring delay of the establishment of a session until receipt of the companion ICMP Echo Request may lead to some user experience degradation.
- o Because of the presence of load-balancers in the path, the ultimate server receiving the SYN packet may not be the one which receives the ICMP message conveying the HOST_ID.
- o Because of the presence of load-balancers in the path, the port number assigned by address sharing may be lost. Therefore the mapping information conveyed in the ICMP may not be sufficient to associate a SYN packet with a received ICMP.
- o The proposal is not compatible with the presence of cascaded NAT. The main reason is each NAT in the path will generate an ICMP message to reveal the internal host identifier. Because these messages will be translated by the downstream address-sharing devices, the remote server will receive multiple ICMP messages and will need to decide which host identifier to use.
- o The ICMP proposal will add traffic overhead for both the server and the address-sharing device.
- o The ICMP proposal is similar to other mechanisms (e.g., Syslog [RFC5424], IPFIX [RFC5101]) for reporting dynamic mappings to a mediation platform (mainly for legal traceability purposes). Performance degradation is likely to be experienced by address-sharing functions because ICMP messages are sent for each new instantiated mapping (and also even if the mapping exists).

o In some scenarios (e.g., Section 3 of [I-D.boucadair-pcp-nat-reveal]), HOST_ID should be interpreted by intermediate devices which embed Policy Enforcement Points (PEP, [RFC2753]) responsible for granting access to some services. These PEPs need to inspect all received packets in order to find the companion (traffic) messages to be correlated with ICMP messages conveying HOST_IDs. This induces more complexity to these intermediate devices.

4.9. Use Out-of-Band Mechanisms (e.g., IDENT)

4.9.1. Description

Another alternative is to retrieve the HOST_ID using a dedicated query channel.

An implementation example may rely on the Identification Protocol (IDENT, [RFC1413]). This solution assumes the address-sharing function implements the server part of IDENT, while remote servers implement the client part of the protocol. IDENT needs to be updated (see [IDENT_NAT]) to be able to return a host identifier instead of the user-id as defined in [RFC1413]. The IDENT response syntax uses the same USERID field described in [RFC1413] but rather than returning a username, a host identifier (e.g., a 16-bit value) is returned [IDENT_NAT]. For any new incoming connection, the server contacts the IDENT server to retrieve the associated identifier. During that phase, the connection may be delayed.

4.9.2. Analysis

o IDENT is specific to TCP. Alternative out-of-band mechanisms may be designed to cover other transport protocols such as UDP.
o This solution requires the address-sharing function to embed an IDENT server.
o A hint should be provided to the ultimate server (or intermediate nodes) that the address-sharing function implements the IDENT protocol. A solution example is to publish this capability using DNS; other solutions can be envisaged.
o An out-of-band mechanism may require some administrative setup (e.g., contract agreement) between the entity managing the address-sharing function and the entity managing the remote server. Such a deployment is not feasible in the Internet at large because establishing and maintaining agreements between ISPs and all service actors is heavy and not scalable.
o Implementations requiring delay of the establishment of a session until receipt of the companion IDENT response may lead to some user experience degradation.

- o The IDENT proposal will add traffic overhead for both the server and the address-sharing device.
- o Performance degradation is likely to be experienced by address-sharing functions embedding the IDENT server. This is further exacerbated if the address-sharing function has to handle an IDENT query for each new instantiated mapping (and also even if the mapping exists).
- o In some scenarios (e.g., Section 3 of [I-D.boucadair-pcp-nat-reveal]), HOST_ID should be interpreted by intermediate devices which embed Policy Enforcement Points (PEP, [RFC2753]) responsible for granting access to some services. These PEPs need to inspect all received packets in order to generate the companion IDENT queries. This may induce more complexity to these intermediate devices.
- o IDENT queries may be generated by illegitimate TCP servers. This would require the address-sharing function to enforce some policies (e.g., rate limit queries, filter based on the source IP address, etc.).

5. Solutions Analysis: Synthesis

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

- o "Success ratio" indicates the ratio of successful communications with remote servers when the HOST_ID is injected using a candidate solution.
- 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-ID	IP Option	TCP Option	HTTP Header	PROXY	Port Set	HIP	ICMP	IDENT
UDP	Yes	Yes	No	No	No	Yes		Yes	No
TCP	Yes	Yes	Yes	No	Yes	Yes		Yes	Yes
HTTP	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes
Encrypted Traffic	Yes	Yes	Yes	No	Yes	Yes		Yes	Yes

Success Ratio	100%	30%	99%	100%	Low	100%	Low	~100%	~100%
Possible Perf Impact	Low to Med	High	Low to Med	Med to High	High	No	N/A	High	High
OS TCP/IP Modif	Yes	Yes	Yes	No	No	No		Yes	Yes
Deployable Today	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes
Notes	(1) (7)	(8)	(8)	(2)	(8)	(1) (3) (7)	(4) (7)	(6) (8)	(1) (6) (8)

Notes:

(1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record).

(2) This solution is widely deployed (e.g., HTTP Severs, Load-Balancers, etc.).

(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. If the client and the server are HIP-enabled, the address-sharing function does not need to insert an identifier. 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.

(6) The solution is inefficient in some scenarios (see Section 5)

(7) The solution is a theoretical construct.

(8) The solution is a documented proposal.

Table 1: Summary of analyzed solutions.

Provided success ratio figures for TCP and IP options are inspired from the results documented in [Options], [I-D.abdo-hostid-tcpopt-implementation], [ExtendTCP].

The provided success ratio for IP-ID is theoretical; it assumes the address-sharing function follows the rules in [RFC6864] to re-write the IP Identification field.

Since PROXY and HIP are not widely deployed, the success ratio for establishing a communication with remote servers using these protocols is low.

The success ratio for the ICMP-based solution is implementation-specific but it is likely to be close to 100%. The success ratio depends on how efficient the solution is implemented on the server side. A remote server which does not support the ICMP-based solution will ignore received companion ICMP messages. An upgraded server will need to delay accepting a session until receiving the companion ICMP message.

The success ratio for IDENT solution is implementation-specific but it is likely to be close to 100%. The success ratio depends on how efficient the solution is implemented on the server side. A remote server which does not support IDENT will accept a session establishment request following its normal operation. An upgraded server will need to delay accepting a session until receipt of the response to the IDENT request it will send to the host.

6. IANA Considerations

This document does not require any action from IANA.

7. Security Considerations

The same security concerns apply for the injection of an IP option, TCP Option and application-related content (e.g., Forwarded HTTP header) 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" HOST_ID (e.g., original IP address).

HOST_ID may be used to leak information about the internal structure of a network behind an address-sharing function. If this behavior is undesired for the network administrator, the address-sharing function can be configured to strip any existing HOST_ID in received packets from internal hosts.

HOST_ID specification documents should elaborate further on threats inherent to each individual solution used to convey the HOST_ID (e.g., use of the IP-ID field to count hosts behind a NAT [Count]).

8. Acknowledgments

Many thanks to D. Wing, C. Jacquenet, J. Halpern, B. Haberman, and P. Yee 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.

The privacy text was provided by A. Cooper.

9. References

9.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[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.

9.2. Informative References

[Count] , "A technique for counting NATted hosts", , <http://www.cs.columbia.edu/~smb/papers/fnat.pdf>.

[ExtendTCP] Honda, M., Nishida, Y., Raiciu, C., Greenhalgh, A., Handley, M. and H. Tokuda,, "Is it still possible to extend TCP?", November 2011, <http://nrg.cs.ucl.ac.uk/mjh/tmp/mboxes.pdf>.

[I-D.abdo-hostid-tcpopt-implementation] Abdo, E., Boucadair, M., and J. Queiroz, "HOST_ID TCP Options: Implementation & Preliminary Test Results", draft-abdo-hostid-tcpopt-implementation-03 (work in progress), July 2012.

[I-D.boucadair-pcp-nat-reveal] Boucadair, M., Reddy, T., Patil, P., and D. Wing, "Using PCP to Reveal a Host behind NAT", draft-boucadair-pcp-nat-reveal-00 (work in progress), November 2012.

[I-D.chen-intarea-v4-uid-header-option] Wu, Y., Ji, H., Chen, Q., and T. ZOU), "IPv4 Header Option For User Identification In CGN Scenario", draft-chen-intarea-v4-uid-header-option-00 (work in progress), March 2011.

[I-D.iab-privacy-considerations]
Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", draft-iab-privacy-considerations-03 (work in progress), July 2012.

[I-D.ietf-appsawg-http-forwarded]
Petersson, A. and M. Nilsson, "Forwarded HTTP Extension", draft-ietf-appsawg-http-forwarded-10 (work in progress), October 2012.

[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-03 (work in progress), December 2011.

[I-D.yourtchenko-nat-reveal-ping]
Yourtchenko, A., "Revealing hosts sharing an IP address using ICMP Echo Request", draft-yourtchenko-nat-reveal-ping-00 (work in progress), March 2012.

[IDENT_NAT]
Wing, D., "Using the Identification Protocol with an Address Sharing Device", August 2012, <draft-wing-intarea-ident>.

[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>.

[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>.

[Proxy] Tarreau, W., "The PROXY protocol", November 2010, <http://haproxy.1wt.eu/download/1.5/doc/proxy-protocol.txt>.

[RFC1413] St. Johns, M.C., "Identification Protocol", RFC 1413, February 1993.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[RFC5101] Claise, B., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.

[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.

[RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, June 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

[RFC6462] Cooper, A., "Report from the Internet Privacy Workshop", RFC 6462, January 2012.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, February 2013.

[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.com

Joe Touch
USC/ISI

Email: touch@isi.edu

Pierre Levis
France Telecom
Caen 14000
France

Email: pierre.levis@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com