

Independent Submission  
Request for Comments: 7768  
Category: Informational  
ISSN: 2070-1721

T. Tsou  
Philips Lighting  
W. Li  
China Telecom  
T. Taylor  
J. Huang  
Huawei Technologies  
January 2016

## Port Management to Reduce Logging in Large-Scale NATs

### Abstract

Various IPv6 transition strategies require the introduction of large-scale NATs (e.g., AFTR and NAT64) to share the limited supply of IPv4 addresses available in the network until transition is complete. There has recently been debate over how to manage the sharing of ports between different subscribers sharing the same IPv4 address. One factor in the discussion is the operational requirement to log the assignment of transport addresses to subscribers. It has been argued that dynamic assignment of individual ports between subscribers requires the generation of an excessive volume of logs. This document suggests a way to achieve dynamic port sharing while keeping log volumes low.

### Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7768>.

## Copyright Notice

Copyright (c) 2016 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.

## Table of Contents

1. Introduction . . . . .	2
2. A Suggested Solution . . . . .	3
3. Issues Of Traceability . . . . .	4
4. Other Considerations . . . . .	5
5. Security Considerations . . . . .	6
6. Informative References . . . . .	7
Appendix A. Configure Server Software to Log Source Port . . . . .	9
A.1. Apache . . . . .	9
A.2. Postfix . . . . .	9
A.3. Sendmail . . . . .	9
A.4. sshd . . . . .	10
A.5. Cyrus IMAP and UW IMAP . . . . .	10
Acknowledgements . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

During the IPv6 transition period, some large-scale NAT devices may be introduced, e.g., Dual-Stack Lite (DS-Lite), Address Family Transition Router (AFTR), and NAT64. When a NAT device needs to set up a new connection for a given internal address behind the NAT, it needs to create a new mapping entry for the new connection, which will contain source IP address, source port or ICMP identifier, converted source IP address, converted source port, protocol (TCP/UDP), etc.

Due to legislation and law enforcement requirement, sometimes it is necessary to log these mappings for a period of time, such as 6 months. The mapping information is highly privacy sensitive; if possible, the information should be deleted as soon as possible. Some high-performance NAT devices may need to create a large amount of new sessions per second. If logs are generated for each mapping entry, the log traffic could reach tens of megabytes per second or more, which would be a problem for log generation, transmission and

storage. According to a test discussed in "Analysis of NAT64 Port Allocation Methods for Shared IPv4 Addresses" [ALLOC-METHODS], in a network with 20,000 subscribers, over a 60-day period, the raw log size can reach 42.5 TB if it is based on per-session log, while the log size will be 40.6 GB if it is based on port blocks. Although compression technologies can be used before log storage, the log size is still big.

[RFC6888], REQ-13 suggests "maximize port utilization" and REQ-14 suggests "minimize log volume". However, it is difficult to achieve both; there will be a trade-off between the efficiency with which ports are used and the rate of generation of log records.

## 2. A Suggested Solution

This document proposes a solution that allows dynamic sharing of port ranges between users while minimizing the number of logs that have to be generated. Briefly, ports are allocated to the user in blocks. Logs are generated only when blocks are allocated or deallocated. This provides the necessary traceability while reducing log generation by a factor equal to the block size, as compared with fully dynamic port allocation.

This is how the proposal works in greater detail. When the user sends out the first packet, a port resource pool is allocated for the user, e.g., assigning ports 2001~2300 of a public IP address to the user's resource pool. Only one log should be generated for this port block. When the NAT needs to set up a new mapping entry for the user, it can use a port in the user's resource pool and the corresponding public IP address. If the user needs more port resources, the NAT can allocate another port block, e.g., ports 3501~3800, to the user's resource pool. Again, just one log needs to be generated for this port block.

[RFC6431] takes this idea further by allocating non-contiguous sets of ports using a pseudorandom function. Scattering the allocated ports in this way provides a modest barrier to port guessing attacks. The use of randomization is discussed further in Section 5.

Suppose now that a given internal address has been assigned more than one block of ports. The individual sessions using ports within a port block will start and end at different times. If no ports in some port block are used for some configurable time, the NAT can remove the port block from the resource pool allocated to a given internal address and make it available for other users. In theory, it is unnecessary to log deallocations of blocks of ports, because

the ports in deallocated blocks will not be used again until the blocks are reallocated. However, the deallocation may be logged when it occurs adding robustness to troubleshooting or other procedures.

The deallocation procedure presents a number of difficulties in practice. The first problem is the choice of timeout value for the block. If idle timers are applied for the individual mappings (sessions) within the block, and these conform to the recommendations for NAT behavior for the protocol concerned, then the additional time that might be configured as a guard for the block as a whole need not be more than a few minutes. The block timer in this case serves only as a slightly more conservative extension of the individual session idle timers. If, instead, a single idle timer is used for the whole block, it must itself conform to the recommendations for the protocol with which that block of ports is associated. For example, REQ-5 of [RFC5382] requires an idle timer expiry duration of at least 2 hours and 4 minutes for TCP.

The next issue with port block deallocation is the conflict between the desire to randomize port allocation and the desire to make unused resources available to other internal addresses. As mentioned above, ideally port selection will take place over the entire set of blocks allocated to the internal address. However, taken to its fullest extent, such a policy will minimize the probability that all ports in any given block are idle long enough for it to be released.

As an alternative, it is suggested that when choosing which block to select a port from, the NAT should omit from its range of choice the block that has been idle the longest, unless no ports are available in any of the other blocks. The expression "block that has been idle the longest" designates the block in which the time since the last packet was observed in any of its sessions, in either direction, is earlier than the corresponding time in any of the other blocks assigned to that internal address.

### 3. Issues Of Traceability

Section 12 of [RFC6269] provides a good discussion of the traceability issue. Complete traceability given the NAT-logging practices proposed in this document requires that the remote destination record the source port of a request along with the source address (and presumably protocol, if not implicit). In addition, the logs at each end must be timestamped, and the clocks must be synchronized within a certain degree of accuracy. Here is one reason for the guard timing on block release, to increase the tolerable level of clock skew between the two ends.

The ability to configure various server applications to record source ports has been investigated, with the following results:

- o Source-port recording can be configured in Apache, Postfix, sendmail, and sshd. Please refer to the Appendix for the configuration guide.
- o Source-port recording is not supported by IIS, Cyrus IMAP, and UW IMAP. But it should not be too difficult to get Cyrus IMAP and UW IMAP to support it by modifying the source code.

Where source-port logging can be enabled, this memo strongly urges the operators to do so. Similarly, intrusion detection systems should capture source port as well as source address of suspect packets.

In some cases [RFC6269], a server may not record the source port of a connection. To allow traceability, the NAT device needs to record the destination IP address of a connection. As [RFC6269] points out, this will provide an incomplete solution to the issue of traceability because multiple users of the same shared public IP address may access the service at the same time. From the point of view of this document, in such situations the game is lost, so to speak, and port allocation at the NAT might as well be completely dynamic.

The final possibility to consider is where the NAT does not do per-session logging even given the possibility that the remote end is failing to capture source ports. In that case, the port allocation policy proposed in this document can be used. The impact on traceability is that analysis of the logs would yield only the list of all internal addresses mapped to a given public address during the period of time concerned. This has an impact on privacy as well as traceability, depending on the follow-up actions taken.

#### 4. Other Considerations

[RFC6269] notes several issues introduced by the use of dynamic, as opposed to static, port assignment. For example, Section 13.2 of that document notes the effect on authentication procedures. These issues must be resolved, but are not specific to the port allocation policy described in this document.

## 5. Security Considerations

The discussion that follows addresses an issue that is particularly relevant to the proposal made in this document. The security considerations applicable to NAT operation for various protocols as documented in, for example, [RFC4787] and [RFC5382] also apply to this proposal.

[RFC6056] summarizes the TCP port-guessing attack, by means of which an attacker can hijack one end of a TCP connection. One mitigating measure is to make the source port number used for a TCP connection less predictable. [RFC6056] provides various algorithms for this purpose.

As Section 3.1 of that RFC notes: "...provided adequate algorithms are in use, the larger the range from which ephemeral ports are selected, the smaller the chances of an attacker are to guess the selected port number." Conversely, the reduced range sizes proposed by the present document increase the attacker's chances of guessing correctly. This result cannot be totally avoided. However, mitigating measures to improve this situation can be taken both at port-block assignment time and when selecting individual ports from the blocks that have been allocated to a given user.

At assignment time, one possibility is to assign ports as non-contiguous sets of values as proposed in [RFC6431]. However, this approach creates a lot of complexity for operations, and the pseudorandomization can create uncertainty when the accuracy of logs is important to protect someone's life or liberty.

Alternatively, the NAT can assign blocks of contiguous ports. However, at assignment time, the NAT could attempt to randomize its choice of which of the available idle blocks it would assign to a given user. This strategy has to be traded-off against the desirability of minimizing the chance of conflict between what [RFC6056] calls "transport protocol instances" by assigning the most-idle block, as suggested in Section 2. A compromise policy might be to assign blocks only if they have been idle for a certain amount of time, and select pseudorandomly between the blocks available according to this criterion. In this case, it is suggested that the time value used be greater than the guard timing mentioned in Section 2, and that no block should ever be reassigned until it has been idle at least for the duration given by the guard timer.

While the block assignment strategy can provide some mitigation of the port-guessing attack, the largest contribution will come from pseudorandomization at port-selection time. [RFC6056] provides a number of algorithms for achieving this pseudorandomization. When

the available ports are contained in blocks, which are not in general consecutive, the algorithms clearly need some adaptation. The task is complicated by the fact that the number of blocks allocated to the user may vary over time. Adaptation is left as an exercise for the implementor.

## 6. Informative References

### [ALLOC-METHODS]

Chen, G., Li, W., Tsou, T., Huang, J., Taylor, T., and J. Tremblay, "Analysis of NAT64 Port Allocation Methods for Shared IPv4 Addresses", Work in Progress, draft-ietf-sunset4-nat64-port-allocation-02, January 2016.

### [APACHE\_LOG\_CONFIG]

The Apache Software Foundation, "Apache Module mod\_log\_config", <[http://httpd.apache.org/docs/2.4/mod/mod\\_log\\_config.html](http://httpd.apache.org/docs/2.4/mod/mod_log_config.html)>.

### [POSTFIX\_LOG\_CONFIG]

"Postfix Configuration Parameters", <<http://www.postfix.org/postconf.5.html>>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <<http://www.rfc-editor.org/info/rfc4787>>.

[RFC5382] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, DOI 10.17487/RFC5382, October 2008, <<http://www.rfc-editor.org/info/rfc5382>>.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <<http://www.rfc-editor.org/info/rfc6056>>.

[RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<http://www.rfc-editor.org/info/rfc6269>>.

[RFC6431] Boucadair, M., Levis, P., Bajko, G., Savolainen, T., and T. Tsou, "Huawei Port Range Configuration Options for PPP IP Control Protocol (IPCP)", RFC 6431, DOI 10.17487/RFC6431, November 2011, <<http://www.rfc-editor.org/info/rfc6431>>.

[RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<http://www.rfc-editor.org/info/rfc6888>>.

[SENDMAIL\_LOG\_CONFIG] O'Reilly, "Sendmail, 3rd Edition, Page 798", December 2002.

[SSHD\_LOG\_CONFIG] "sshd\_config OpenSSH SSH daemon configuration file", <[http://www.openbsd.org/cgi-bin/man.cgi?query=sshd\\_config&sektion=5](http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config&sektion=5)>.



## Appendix A. Configure Server Software to Log Source Port

### A.1. Apache

The user can use the `LogFormat` command to define a customized log format and use the `CustomLog` command to apply that log format. `"%a"` and `"%{remote}p"` can be used in the format string to require logging the client's IP address and source port, respectively. This feature has been available since Apache version 2.1.

A detailed configuration guide can be found at `[APACHE_LOG_CONFIG]`.

### A.2. Postfix

In order to log the client source port, macro `smtpd_client_port_logging` should be set to "yes" in the configuration file `[POSTFIX_LOG_CONFIG]`.

This feature has been available since Postfix version 2.5.

### A.3. Sendmail

Sendmail has a macro `#{client_port}` storing the client port. To log the source port, the user can define some check rules. Here is an example that should be in the `.mc` configuration macro `[SENDMAIL_LOG_CONFIG]`:

```
LOCAL_CONFIG
Klog syslog
```

```
LOCAL_RULESETS
SLocal_check_mail
R $* $@ $(log Port_Stat $&{client_addr} $&{client_port} $)
```

This feature has been available since version 8.10.

#### A.4. sshd

SSHD\_CONFIG(5) OpenBSD Programmer's Manual SSHD\_CONFIG(5) NAME  
 sshd\_config - OpenSSH SSH daemon configuration file LogLevel Gives the verbosity level that is used when logging messages from sshd(8). The possible values are: QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2, and DEBUG3. The default is INFO. DEBUG and DEBUG1 are equivalent. DEBUG2 and DEBUG3 each specify higher levels of debugging output. Logging with a DEBUG level violates the privacy of users and is not recommended. SyslogFacility Gives the facility code that is used when logging messages from sshd(8). The possible values are: DAEMON, USER, AUTH, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, and LOCAL7. The default is AUTH.

sshd supports logging the client IP address and client port when a client starts connection since version 1.2.2; here is the source code in sshd.c:

```
...
verbose("Connection from %.500s port %d", remote_ip, remote_port);
...
```

sshd supports logging the client IP address when a client disconnects in version 1.2.2 to version 5.0. Since version 5.1, sshd supports logging the client IP address and source port. Here is the source code in sshd.c:

```
...
/* from version 1.2.2 to 5.0*/
verbose("Closing connection to %.100s", remote_ip);
...

/* since version 5.1*/
verbose("Closing connection to %.500s port %d",
remote_ip, remote_port);
```

In order to log the source port, the LogLevel should be set to VERBOSE [SSHD\_LOG\_CONFIG] in the configuration file:

```
LogLevel      VERBOSE
```

#### A.5. Cyrus IMAP and UW IMAP

Cyrus IMAP and UW IMAP do not support logging the source port for the time being. Both software use syslog to create logs; it should not be too difficult to get it supported by adding some new code.

## Acknowledgements

Mohamed Boucadair reviewed the initial document and provided useful comments to improve it. Reinaldo Penno, Joel Jaeggli, and Dan Wing provided comments on the subsequent draft version that resulted in major revisions. Serafim Petsis provided encouragement to publish the document after a hiatus of two years.

The authors are grateful to Dan Wing for his help in moving this document forward, and in particular for his helpful comments on its content.

## Authors' Addresses

Tina Tsou  
Philips Lighting  
3 Burlington Woods Dr #4t  
Burlington, MA 01803  
United States

Email: [tina.tsou@philips.com](mailto:tina.tsou@philips.com)

Weibo Li  
China Telecom  
109, Zhongshan Ave. West, Tianhe District  
Guangzhou 510630  
P.R. China

Email: [mweiboli@gmail.com](mailto:mweiboli@gmail.com)

Tom Taylor  
Huawei Technologies  
Ottawa  
Canada

Email: [tom.taylor.stds@gmail.com](mailto:tom.taylor.stds@gmail.com)

James Huang  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: [James.huang@huawei.com](mailto:James.huang@huawei.com)