Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: October 13, 2018

D. O'Reilly

April 11, 2018

Approaches to Address the Availability of Information in Criminal Investigations Involving Large-Scale IP Address Sharing Technologies
draft-daveor-cgn-logging-04

Abstract

The use of large-scale IP address sharing technologies (commonly known as "Carrier-Grade NAT" and "A+P") presents a challenge for law enforcement agencies due to the fact that incoming source port information is not routinely logged by Internet-facing servers. The absence of this information means that it is becoming increasingly difficult for law enforcement agencies to identify suspects in criminal activity online. This document considers the reasons why source port information is not routinely logged by Internet-facing servers and makes recommendations to help improve the situation. A deployment maturity model has been developed and a study of the support for logging incoming source port information in common server software is also presented.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 2
2. Scope 4
3. Centralised Connection Logging 5
4. Challenges to Capturing Source Port 7
4.1. Lack of Awareness 7
4.2. Lack of Support for Logging Source Port 8
4.3. Additional Storage Requirements 8
4.4. Default Log Formats 8
4.5. Breaking Existing Tooling 9
4.6. Accuracy of Recorded Time 9
4.7. Translation of Source Port by Endpoint Infrastructure . . 9
5. Comparison Model 10
6. Support for Logging Source Port 10
7. Recommendations 11
7.1. Raise Awareness of the Importance of Logging Source Port 12
7.2. Increase Support for Logging Source Port 12
7.3. Update Default Log Formats 12
7.4. Adequate Timestamp Accuracy in Logs 13
7.5. Source Port Translation in Endpoint Infrastructure . . . 13
8. IANA Considerations 14
9. Security Considerations 14
10. Acknowledgements 15
11. References 15
11.1. Informative References 15
11.2. Normative References 15
Appendix A. Support for Source Port Logging in Various Server Software 17
Author's Address 18

1. Introduction

Large-scale IP address sharing technologies (such as "Carrier-Grade NAT", [RFC6888]) are a helpful tool for extending the life of IPv4 addresses by allowing multiple endpoints to share a small number of IPv4 addresses. A related category of technologies, known as "Address plus Port", or "A+P" [RFC6346], are also used for large-

scale IP address sharing, achieved in these cases by using some of the port number bits for addressing purposes. A number of such technologies have been discussed and deployed, such as Dual-Stack Lite [RFC6333], NAT64 [RFC6146], NAT444 [I-D.shirasaki-nat444], Lightweight 4over6 [RFC7596], MAP-E [RFC7597] and MAP-T [RFC7599].

All of these technologies involve extending the space of available IPv4 addresses by mapping communication from multiple endpoints to a single, or small number of shared addresses, through the use of port numbers. The detail of how this is achieved in each technology varies, but the principle remains the same in all cases.

From the perspective of a server on the Internet, endpoint traffic that has passed through IP address sharing infrastructure appears to be originating from the IP address of the address sharing appliance. Common practice at the present time is for servers to log the connection time and source IP address of incoming connections. However, the IP address of the address sharing appliance is not sufficient to identify the true source of the traffic because potentially hundreds or thousands of individual endpoints were using that IP address at the same time. If the need arises during a criminal investigation to identify the source of a specific connection, the source port and exact connection time will also be required. Without this additional information it is highly unlikely that it will be possible for law enforcement authorities to progress their investigations.

Information is required from at least two sources to establish the link from the logs of an Internet-facing server to a specific subscriber endpoint:

1. The administrator of the Internet-facing server must have logged enough information to enable the operator of the IP address sharing infrastructure to isolate a specific subscriber endpoint.

2. The operator of the IP address sharing infrastructure must have logged sufficient information (for a sufficient length of time) to be able, when provided with adequate data by a law enforcement agency, to isolate the relevant subscriber endpoint.

The operators of large-scale IP address sharing infrastructure, typically Internet Service Providers, are usually required by law to maintain records of which endpoint was using a particular IP address and port at a particular time. The period of time for which these records must be retained is defined by national legislation. Irrespective of whether (and for how long) these records are available, a starting point is needed to indicate to an investigating law enforcement agency that a particular endpoint was involved in a

suspected criminal activity under investigation. Without such a starting point, it would be very difficult to progress the investigation even as far as engagement with the operator of the address sharing infrastructure. The records of Internet-facing servers are often a crucial source of this type of evidence.

It has been recognised for some time that IP address sharing presents a challenge to the ability to trace network use and abuse [RFC7620]. Further, it has also been recognised that this challenge is likely to become more severe and widespread with the increased use of large-scale address sharing [RFC6269]. More recently, Europol has highlighted the issue of large-scale IP address sharing as a threat to Internet governance [EUROPOL_IOCTA]. It is reported that the problem of crime attribution related to the use of carrier-grade NAT technologies is regularly encountered by 90% of respondents to a survey on the topic.

Address sharing, including large-scale address sharing, is required as long as the use of IPv4 continues. Full deployment of IPv6 has the potential to ultimately eliminate the current attribution issues arising from the use of large-scale address sharing technologies, although presumably new attribution challenges will arise in that scenario. Since it is impossible to anticipate if or when full migration to IPv6 will take place, it is prudent to consider the implications of the transitionary technologies until the need for them has been eliminated.

2. Scope

Previous work has already suggested as best practice the logging by Internet-facing servers of source IP address, source port and exact connection time [RFC6302]. However, this continues to be exceptional, rather than routine, logging practice. The purpose of this document is to consider in more detail how it might be possible to bring about routine logging by Internet-facing servers of the information needed to re-establish the ability to trace network abuse for criminal investigative purposes. This document specifically does not address or consider the logging requirements of operators of large-scale address sharing infrastructre. Instead, the focus is on the logging considerations of operators of Internet-facing servers. The main contributions of this document are:

1. To consider the reasons why source port logging is not routinely carried out.

2. To identify some possible solutions and workarounds for the reasons that source port logging is not routinely carried out.

3. To examine the feasibility of source port logging from the perspective of software support for this feature.

Clearly no single solution will address the problem of crime attribution on the Internet. Load balancers, proxies and other network infrastructure may also, intentionally or as a side-effect, obfuscate the true source of Internet traffic and these problems will continue to exist with or without the presence of large-scale address sharing technologies (like Carrier-Grade NAT and A+P). Nevertheless, at the time of writing large-scale address sharing technologies present a significant challenge to crime attribution, as highlighted by Europol in the above referenced link, and this document attempts to consider the challenges specifically presented by that category of technologies.

The discussion begins by considering whether centralised connection logging is a viable solution to the problem of subscriber identification in criminal investigations. This is followed by an examination of the reasons why source port logging is not currently routinely carried out. A model has been developed for the comparison of the maturity of various server deployments to log source port and a study of common server software has been performed to assess the status of support for this functionality. Many, but not all, enterprise server solutions that were examined made the logging of source port either "Possible" or "Feasible", as defined in the maturity model. Only one type of server software examined made the logging of source port "Default".

3. Centralised Connection Logging

When large-scale IP address sharing technologies are used, source IP address is no longer a sufficient identifier of an individual subscriber. At a minimum, source port and accurate timestamp information are also required to distinguish between the potentially large number of individual users of a specific IP address at a particular time. [RFC6269] points out that there are two solutions to the question of how adequate information can be recorded to identify the parties to a particular connection. They are:

1. Operators of IP address sharing infrastructure log mappings between (source IP address, source port) combinations and their subscribers. Server operators log the IP address and source port of incoming connections. This is referred to as source port logging.

2. Instead of relying on server operators to log the source port of incoming connections, operators of IP address sharing infrastructure log all combinations of (external IP address,

external port, destination IP address) for outgoing connections. This is referred to as connection logging. Server operators log the IP address and timestamp of incoming connections, which is the common current practice.

Two challenges to the use of connection logging by operators of IP address sharing infrastructure are also presented in RFC6269. Briefly:

o The volumes of data involved make centralised recording of destination IP addresses infeasible.

o Many individuals using the same IP address to access a popular destination (e.g. a popular website) might mean that it is not possible to distinguish between the activity of one subscriber and another, even if connection records are kept by the operator of the address sharing infrastructure.

The first issue raised is that the volumes of data involved make centralised recording of destination IP addresses infeasible. Whether destination IP addresses are recorded or not, the volume of logs generated by a large-scale IP address sharing infrastructure will be substantial, and some approaches have been proposed to address this hurdle and make central connection logging more feasible, such as deterministic allocation of ports [RFC6269],[RFC7422] or allocation of port ranges [RFC7768], [RFC6346]. While arguments of infeasibility are not arguments in principle why such logging cannot be done, the volumes of data involved in recording every single outgoing connection in a large Internet service provider represent legitimate technical, commercial and operational arguments for why it can not work in practice. Some representative figures for the scales of data involved can be found in [RFC7422], wherein it is estimated that the logging overhead would be of the order of 150MB per subscriber, per month. For a service provider with one million subscribers, this would produce a volume of logs (uncompressed) of the order of 150 terabytes per month. Aside from the technical overhead of storing such a volume of data, searching and locating relevant records over an extended, legally mandated retention period would also present a significant technical challenge.

The second point raised in [RFC6269] against connection logging by operators of IP address sharing infrastructure suggests that even if connection logs store all combinations of (timestamp, source IP, source port, destination IP), if this information is queried in the absence of source port because source port has not been recorded by the destination IP, this would not be sufficient to distinguish the activity of one individual from another in cases where the

destination IP is a popular one. This problem is further exacerbated in the case of protocols that make multiple connections per session (e.g. HTTP/HTTPS). The implication of this point is that connection logging, despite potential significant technical and operational overhead, cannot guarantee that the information retained is sufficient to identify an individual suspect, even when all required records are available.

Finally, the privacy concerns arising from connection logging in this scenario have been repeatedly raised [RFC6888] and [I-D.ietf-behave-ipfix-nat-logging].

In summary, it is certainly clear that operators of address sharing infrastructure need to retain records to enable the identification of suspects, and such records must consist of, at least, sufficient information to identify an individual subscriber when provided with a timestamp, source IP, source port and destination IP. However, there is no centralised solution available that removes the need for server operators to retain source port information.

4. Challenges to Capturing Source Port

It is relatively easy to articulate the reason why the operator of an Internet-facing server would wish to retain source port information for incoming connections. If the server operator (or the users that they serve) finds themselves the victim of a crime, it is preferable that all information that could be needed by the server operator to facilitate a criminal investigation is available. On the other hand, there are reasons why a server operator might not have the required source port information. This section enumerates the factors that could negatively influence both the ability and the inclination of server operators to capture and record source port information.

4.1. Lack of Awareness

Server operators are principally focussed on delivering the services for which they are operating their infrastructure. One of the main problems with the increasing use of IP address sharing technologies is the lack of awareness on the part of server operators that there are direct implications for them in case they should become the victim of a crime.

At the time of writing, a minimal amount of material is available online concerning this issue, even for those actively seeking to find out about source port logging. Where specific guidance or information has been provided by vendors in relation to the configuration of source port logging, no explanation is provided for

why this might be something that server operators might consider desirable. For example [MSDN_IIS_LOG].

There is, therefore, a considerable awareness gap between the importance of this issue for the purpose of investigating criminal activity online and the awareness of those who need to act in advance of any criminality taking place to ensure that the information needed to facilitate a future investigation is available.

4.2. Lack of Support for Logging Source Port

Before a server operator can decide to log source port information, the server software must support logging of the source port of incoming connections. Many, but not all major software distributions support the logging of the source port of incoming connections. Clearly lack of support in server software is a technical obstacle for a server operator to logging source port at the endpoint. It may still be possible to log source port at some location before the server endpoint (e.g. at a reverse proxy) but absence of support in server software will mean that endpoint logging will not be possible.

4.3. Additional Storage Requirements

In cases where it is possible to simply add source port to the list of fields recorded in log entries, the additional storage required to preserve source port data is minimal; in the region of six bytes per log entry (maximum of five ASCII digits for the source port plus an additional delimiter).

However, in some cases where software supports logging source port of incoming connections, it has been noted that this can only be achieved by enabling verbose or debug logging in the software. This would substantially (and unnecessarily) increase the size of logs produced by the server and would also, in all probability, reduce the production performance of the server. These factors would undoubtedly negatively influence the decision by a server operator to log incoming source port.

4.4. Default Log Formats

Many major software distributions provide default log formats in their configuration files. A review of the default log format of some common server software has been carried out and in only one case was it found that the source port of incoming connections is logged by any of the default log formats.

4.5. Breaking Existing Tooling

Much commercial and free log analysis software, by default, expects logs to be in a particular format. Consider, for example, the ubiquity of the Apache Common and Extended Log Formats. The software can usually be configured to parse arbitrary log formats, but this is additional configuration work for a server operator. For example: [ANALOG_LOG_CONFIG],[AWSTATS_LOG_CONFIG]. Without migration planning, a change to default log formats would most likely cause substantial disruption to a considerable amount of downstream processing of server log files. In addition to commercially available software, many administrators have developed or downloaded scripts that expect logs to be in a standard log format.

Therefore, log processing software, and in particular custom scripts, may break if default log formats change unexpectedly. At least, the tooling may need to be updated to correctly process the additional fields newly present in log file.

4.6. Accuracy of Recorded Time

As well as recording the IP address and source port of the connection, it is important to record the exact time of the connection. It has been suggested that there is a need for keeping the exact time against some sort of global standard (e.g. NTP) [RFC6302], however this may not be possible for practical, security or legacy reasons. In practice, it is usually not necessary to keep time against a global standard, as long as time is recorded consistently. The reason for this is that any time offset between the server and the time recorded in another organisation's records (running address sharing infrastructure) can be calculated and compensated for manually. Time offsets of this nature are commonly encountered and well understood in the digital forensics world.

4.7. Translation of Source Port by Endpoint Infrastructure

It is common for an incoming connection to terminate somewhere other than the actual server that is ultimately handling the connection. Load balancers, proxies or denial of service countermeasures may be present to improve the efficiency or availability of the platform, any one of which could potentially terminate the incoming connection. The operation of these types of endpoint infrastructure can cause translation of the incoming connection parameters, including source port, before the connection is established to the actual server endpoint.

In such cases the source port logged at the server endpoint is a source port that only has meaning within the endpoint infrastructure

and in most cases will not carry any information about the source port in use at the connection origin, in this case the connection origin being the large-scale address sharing infrastructure. In the worst case scenario (from a crime attribution point of view), the endpoint infrastructure may obfuscate the true source connection information in a way that is unrecoverable.

5. Comparison Model

A model has been developed to assist with comparison of the maturity of server software deployments to store and retrieve source port information for incoming connections. The model is depicted in Figure 1.

```
+--------------------------------------------------------------+
| Possible -> Feasible -> Default -> Manageable -> Accessible |
+--------------------------------------------------------------+
```

Figure 1

- "Possible": Means that the server software supports, in any way, the ability to record source ports for incoming connections.
- "Feasible": Means that it there are no significant performance or storage implications for enabling the storage of source ports.
- "Default": Means that, at a minimum, at least one of the default log formats provided with the software distribution enables the storage of source ports.
- "Manageable": Means that tooling is, or has been, build or adapted to support the storage of source ports.
- "Accessible": Means that it is possible to identify and retrieve relevant records in the stored log data.

6. Support for Logging Source Port

Open-source research has been conducted to assess the status of support for logging of source port information in common server software.

The assessment criteria were as follows:

- Server software is categorised as "Possible" if there was any way identified to cause the logging of source port.

- Server software is categorised as "Feasible" if the logging of source port does not require increasing the log level to cause the logging of source port to be possible. In other words, if a server requires enabling verbose, debug or audit logging in order to be able to record source port then logging is "Possible" but not "Feasible".

- Server software is categorised as "Default" if at least one of the available default log formats enables logging of the incoming source port, or if source port is logged by default.

- The "Manageable" and "Accessible" aspects of the comparison model relate to specific deployments and are therefore not considered in the assessment of server software support.

The latest versions of 16 common server software packages have been examined and documentation has been research to identify if and how source port logging can be enabled. The findings are described in Appendix A. Online documentation has been examined to identify if and how source port logging can be enabled. The results are presented in the following table:

Possible	Feasible	Default	Manageable	Accessible
13	11	1	N/A	N/A

Table 1: Support Table

It was noted that only one of the server software packages examined (OpenSSH version 7.5) enables the logging of incoming source port by default. This conclusion has been reached despite using the most generous possible interpretation of "Default", whereby meeting the criteria for "Default" is achieved when logging of source port is offered as a possible default, rather than requiring that logging of source port is enabled by default. In due course, as awareness of this issue increases, it is envisioned that a stricter interpretation of "Default" would be more appropriate, requiring that the logging of source port be enabled by default.

7. Recommendations

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 [RFC2119].

The recommendations presented below are courses of action that have been identified based on the current state of source port logging and the challenges described above.

7.1. Raise Awareness of the Importance of Logging Source Port

Publishers of both free and commercial software SHOULD release deployment guidance or best practice that describes why server administrators need to record source port information, with instructions for how this can be done. This will help to address the lack of awareness of the importance of this issue.

Considering also the awareness of those who are building software applications, or otherwise involved with coding of Internet-facing applications, secure coding guidance SHOULD be updated to include reference to source port information, particularly where such guidance already touches on the issue of logging. For example the OWASP Secure Coding Practices specifies a list of important log event data [OWASP_SCP]. However the "important log event data" list does not, at the time of writing, include source port.

7.2. Increase Support for Logging Source Port

Many software packages support logging of source port information, but only ten out of the sixteen examined support logging in a way that would not significantly negatively impact the operation of the server software. Software publishers therefore need to consider their level of support of logging source port. In particular, software SHOULD support the logging of source port and SHOULD do so in a way that does not substantially impact on production performance.

7.3. Update Default Log Formats

In cases where a software package has support for logging of incoming source port, the configuration SHOULD incorporate one or more optional log formats that include incoming source port as a field logged by default. Obviously this will not have any impact on deployments of the software that are already in place but for future deployments, the incorporation of source port into "out of the box" log formats will mean that those administrators using unaltered default log formats will automatically store the needed information. Software vendors SHOULD provide a default log format that includes logging of source port, as described in this document.

An alternative approach, taking into account the fact that changes to log formats might break downstream tooling, would be to configuring parallel logging of connection information to a separate log stream.

This would also be a possible solution that could be used by those server software types that log via syslog. In this case, software publishers SHOULD produce guidance on how to configure syslog to log connection information parallel to the main log files. Such a solution would help to ease the transition to an alternate log format since current log formats would not need to be changed because the required source port information is stored separately, but can still be correlated with the main log files if needed.

7.4. Adequate Timestamp Accuracy in Logs

In order to query their records, operators of large-scale address sharing infrastructure will usually need connection times specified with at least the granularity of a second. Consideration should be given by server operators to making sure that the times recorded in their log files have sufficient accuracy to allow identification of the required records. Server software SHOULD be able to log time with at least the granularity of a second.

There are many reasons why it is may not be possible for servers to record logs with reference to a global time source. This could include scenarios should as security sensitive networks, or internal production networks. As long as times are recorded consistently, it should be possible to measure the offset from a traceable global time source (if required) for the purposes of quering records at another source. If the entity controlling the server is aware that there is an offset required to synchronise with a global time source, it is expected that the offset would be indicated by the entity while the logs were being collected.

Adequate timstamp accuracy also needs to be considered by software developers when they are producing software. Although the recording of time is mentioned in the OWASP Secure Coding Practices, the required accuracy/granularity of the recorded time is not discussed [OWASP_SCP]. Development guidance SHOULD include clarifying that times need to be recorded with at least the granularity of a second.

7.5. Source Port Translation in Endpoint Infrastructure

In cases where endpoint infrastructure terminates incoming connections (proxies, load balancers, etc.), and the infrastructure translates incoming source port information, there is a risk that the important crime attribution information may be lost. One possibility is to log source port information at the endpoing infrastructure and this may be an appropriate solution in some cases. However, this may lead to an excessive volume of logging, depending on the particular scenario. For example if the intermediate infrastructure is being used to mitigate DDoS attacks, logging all incoming traffic would

potentially lead to logging of all incoming DDoS connections. This would clearly be an undesirable outcome.

An alternative solution is to pass information about the original connection (before mapping/translation of connection information takes place) to the actual endopint. Solutions to achieve this already exist for certain application layer protocols. The Forwarded HTTP Extention [RFC7239], for example, supports (as an optional feature) the tranfer of source port information in the "Forwarded For" header, and this technique can also support multiple layers of proxying without loss of attribution. Therefore, endpoint infrastructure that translates source ports SHOULD pass the original connection information through to the Internet-facing server for logging purposes.

8. IANA Considerations

This memo includes no request to IANA.

9. Security Considerations

Clearly a balance needs to be struck between individual right to privacy and law enforcement access to data during criminal investigations. On the one hand, the routine logging of any additional information has the potential to introduce risks related to privacy and human rights. On the other hand, there is a societal, crime prevention requirement to address the information gap created by large-scale address sharing technologies. Across the world there are also a broad spectrum of legislative regimes and human rights challenges, interpretation of which relate directly to this question.

IP addresses are routinely logged today and this information can be used for identification of people online in some cases. The cases in which an IP addresses does not identify an individual directly are not necessarily apparent to the person performing the logging (who cannot tell, for example, if the true source of the traffic is behind a NAT or other form of proxy) and the same is true even if source port is logged. It is not apparent that there is any additional risk to individual privacy between the case when a single piece of endpoint identifying information (source IP address) is logged versus the case when two pieces of endpoint identifying information (source IP address and source port) are logged. Balancing this against the significant advantages from the crime attribution point of view suggests that this may be a worthwhile approach.

10. Acknowledgements

Several members of the v6ops mailing list provided valuable feedback and discussion on early drafts of this document. In particular, Tom Herbert, Ca By, Ole Troan, Lee Howard, Erik Nygren, Fred Baker, Fernando Gont, Gert Doering, Mark Smith, Jordi Palet Martinez, DY Kim, Mark Andrews and T. Petch. Special acknowledgement also goes to Mohamed Boucadiar who has provided ongoing feedback throughout the document development process.

11. References

11.1. Informative References

[I-D.ietf-behave-ipfix-nat-logging]
Sivakumar, S. and R. Penno, "IPFIX Information Elements for logging NAT Events", draft-ietf-behave-ipfix-nat-logging-13 (work in progress), January 2017.

[I-D.shirasaki-nat444]
Yamagata, I., Shirasaki, Y., Nakagawa, A., Yamaguchi, J., and H. Ashida, "NAT444", draft-shirasaki-nat444-06 (work in progress), July 2012.

11.2. Normative References

[ANALOG_LOG_CONFIG]
Analog, "Analog 6.0: Log formats", 2017, <http://mirror.reverse.net/pub/analog/docs/logfmt.html>.

[AWSTATS_LOG_CONFIG]
AWStats, "AWStats Installation, Configuration and Reporting (for version 7.6)", 2017, <https://awstats.sourceforge.io/docs/awstats_setup.html>.

[EUROPOL_IOCTA]
Europol, "The Internet Organised Crime Threat Assessment", 2016, <https://www.europol.europa.eu/activities-services/main-reports/internet-organised-crime-threat-assessment-iocta-2016>.

[MSDN_IIS_LOG]
Microsoft, "IIS 8.5 - How to log client port number", 2015, <https://blogs.msdn.microsoft.com/amb/2015/11/12/iis-8-5-how-to-log-client-port-number/>.

[OWASP_SCP]
OWASP, "OWASP Secure Coding Practices Quick Reference Guide", 2010, <https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[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, <https://www.rfc-editor.org/info/rfc6269>.

[RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, DOI 10.17487/RFC6302, June 2011, <https://www.rfc-editor.org/info/rfc6302>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <https://www.rfc-editor.org/info/rfc6333>.

[RFC6346] Bush, R., Ed., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, DOI 10.17487/RFC6346, August 2011, <https://www.rfc-editor.org/info/rfc6346>.

[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, <https://www.rfc-editor.org/info/rfc6888>.

[RFC7239] Petersson, A. and M. Nilsson, "Forwarded HTTP Extension", RFC 7239, DOI 10.17487/RFC7239, June 2014, <https://www.rfc-editor.org/info/rfc7239>.

[RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments", RFC 7422, DOI 10.17487/RFC7422, December 2014, <https://www.rfc-editor.org/info/rfc7422>.

[RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <https://www.rfc-editor.org/info/rfc7596>.

[RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <https://www.rfc-editor.org/info/rfc7597>.

[RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <https://www.rfc-editor.org/info/rfc7599>.

[RFC7620] Boucadair, M., Ed., Chatras, B., Reddy, T., Williams, B., and B. Sarikaya, "Scenarios with Host Identification Complications", RFC 7620, DOI 10.17487/RFC7620, August 2015, <https://www.rfc-editor.org/info/rfc7620>.

[RFC7768] Tsou, T., Li, W., Taylor, T., and J. Huang, "Port Management to Reduce Logging in Large-Scale NATs", RFC 7768, DOI 10.17487/RFC7768, January 2016, <https://www.rfc-editor.org/info/rfc7768>.

Appendix A. Support for Source Port Logging in Various Server Software

The table below enumerates the findings of best-effort, open-source review of documentation of the various products. Where it has been indicated that it is not possible to log source port then either (a) no reference has been identified in online documentation to indicate how source port logging can be enabled, or (b) a reference positively indicating that logging of source port is not possible has been found.

Category	Server	Version	Possible	Feasible	Default
HTTP	Apache HTTPD	2.4.25	Yes	Yes	No
HTTP	IIS	10	Yes	Yes	No
HTTP	Tomcat	8.5.15	Yes	Yes	No
HTTP	Squid	3.5.25	Yes	Yes	No
HTTP	nginx	1.12.0	Yes	Yes	No
Mail	sendmail	8.15.2	Yes	Yes	No
Mail	Microsoft Exchange Server	2016	Yes	No	No
Mail	Postfix	2.10.0	Yes	Yes	No
Mail	Exim	4.89	Yes	Yes	No
Mail	Dovecot	2.2.30.1	Yes	Yes	No
Mail	UW IMAP	imap-2007f	No	No	No
DBase	Oracle	12.2.0.1	No	No	No
DBase	MySQL	5.7.18	No	No	No
DBase	Microsoft SQL Server	2016	Yes	No	No
DBase	PostgreSQL	9.6.3	Yes	Yes	No
SSH	OpenSSHD	7.5	Yes	Yes	Yes

Table 2: Support for Logging Incoming Source Port

Author's Address

David O'Reilly
Ireland

Email: rfc@daveor.com