

Internet Area WG
Internet-Draft
Intended status: Best Current Practice
Expires: December 12, 2018

R. Bonica
Juniper Networks
F. Baker
Unaffiliated
G. Huston
APNIC
R. Hinden
Check Point Software
O. Troan
Cisco
F. Gont
SI6 Networks
June 10, 2018

IP Fragmentation Considered Fragile
draft-bonica-intarea-frag-fragile-02

Abstract

This document provides an overview of IP fragmentation. It explains how IP fragmentation works and why it is required. As part of that explanation, this document also explains how IP fragmentation reduces the reliability of Internet communication.

This document also proposes alternatives to IP fragmentation. Finally, it provides recommendations for application developers and network operators.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 12, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. IP Fragmentation	3
2.1. Links, Paths, MTU and PMTU	3
2.2. Upper-layer Protocols	5
3. Requirements Language	7
4. IP Fragmentation Reduces Reliability	7
4.1. Middle Box Failures	7
4.2. Partial Filtering	8
4.3. Suboptimal Load Balancing	8
4.4. Security Vulnerabilities	9
4.5. Blackholing Due to ICMP Loss	11
4.5.1. Transient Loss	12
4.5.2. Incorrect Implementation of Security Policy	12
4.5.3. Persistent Loss Caused By Anycast	13
4.6. Blackholing Due To Filtering	13
5. Alternatives to IP Fragmentation	14
5.1. Transport Layer Solutions	14
5.2. Application Layer Solutions	15
6. Applications That Rely on IPv6 Fragmentation	16
6.1. DNS	16
6.2. OSPFv3	17
6.3. Packet-in-Packet Encapsulations	17
7. Recommendations	17
7.1. For Application Developers	17
7.2. For Network Operators	17
8. IANA Considerations	18
9. Security Considerations	18
10. Acknowledgements	18
11. References	18
11.1. Normative References	18
11.2. Informative References	19

Appendix A. Contributors' Address	22
Authors' Addresses	22

1. Introduction

Operational experience [RFC7872] [Huston] reveals that IP fragmentation reduces the reliability of Internet communication. This document provides an overview of IP fragmentation. It explains how IP fragmentation works and why it is required. As part of that explanation, this document also explains how IP fragmentation reduces the reliability of Internet communication.

This document also proposes alternatives to IP fragmentation. Finally, it provides recommendations for application developers and network operators.

2. IP Fragmentation

2.1. Links, Paths, MTU and PMTU

An Internet path connects a source node to a destination node. A path can contain links and intermediate systems. If a path contains more than one link, the links are connected in series and an intermediate system connects each link to the next. An intermediate system can be a router or a middle box.

Internet paths are dynamic. Assume that the path from one node to another contains a set of links and intermediate systems. If the network topology changes, that path can also change so that it includes a different set of links and intermediate systems.

Each link is constrained by the number of bytes that it can convey in a single IP packet. This constraint is called the link Maximum Transmission Unit (MTU). IPv4 [RFC0791] requires every link to have an MTU of 68 bytes or greater. IPv6 [RFC8200] requires every link to have an MTU of 1280 bytes or greater. These are called the IPv4 and IPv6 minimum link MTU's.

Each Internet path is constrained by the number of bytes that it can convey in a IP single packet. This constraint is called the Path MTU (PMTU). For any given path, the PMTU is equal to the smallest of its link MTU's. Because Internet paths are dynamic, PMTU is also dynamic.

For reasons described below, source nodes estimate the PMTU between themselves and destination nodes. A source node can produce extremely conservative PMTU estimates in which:

- o The estimate for each IPv4 path is equal to the IPv4 minimum link MTU.
- o The estimate for each IPv6 path is equal to the IPv6 minimum link MTU.

While these conservative estimates are guaranteed to be less than or equal to the actual MTU, they are likely to be much less than the actual PMTU. This may adversely affect upper-layer protocol performance.

By executing Path MTU Discovery (PMTUD) [RFC1191] [RFC8201] procedures, a source node can maintain a less conservative, running estimate of the PMTU between itself and a destination node. According to these procedures, the source node produces an initial PMTU estimate. This initial estimate is equal to the MTU of the first link along the path to the destination node. It can be greater than the actual PMTU.

Having produced an initial PMTU estimate, the source node sends non-fragmentable IP packets to the destination node. If one of these packets is larger than the actual PMTU, a downstream router will not be able to forward the packet through the next link along the path. Therefore, the downstream router drops the packet and sends an Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443] Packet Too Big (PTB) message to the source node. The ICMP PTB message indicates the MTU of the link through which the packet could not be forwarded. The source node uses this information to refine its PMTU estimate.

PMTUD produces a running estimate of the PMTU between a source node and a destination node. Because PMTU is dynamic, at any given time, the PMTU estimate can differ from the actual PMTU. In order to detect PMTU increases, PMTUD occasionally resets the PMTU estimate to the MTU of the first link along path to the destination node. It then repeats the procedure described above.

PMTUD has the following characteristics:

- o It relies on the network's ability to deliver ICMP PTB messages to the source node.
- o It is susceptible to attack because ICMP messages are easily forged [RFC5927].

FOOTNOTE: According to RFC 0791, every IPv4 host must be capable of receiving a packet whose length is equal to 576 bytes. However, the

IPv4 minimum link MTU is not 576. Section 3.2 of RFC 0791 explicitly states that the IPv4 minimum link MTU is 68 bytes.

FOOTNOTE: In the paragraphs above, the term "non-fragmentable packet" is introduced. A non-fragmentable packet can be fragmented at its source. However, it cannot be fragmented by a downstream node. An IPv4 packet whose DF-bit is set to zero is fragmentable. An IPv4 packet whose DF-bit is set to one is non-fragmentable. All IPv6 packets are also non-fragmentable.

FOOTNOTE: In the paragraphs above, the term "ICMP PTB message" is introduced. The ICMP PTB message has two instantiations. In ICMPv4 [RFC0792], the ICMP PTB message is Destination Unreachable message with Code equal to (4) fragmentation needed and DF set. This message was augmented by [RFC1191] to indicate the MTU of the link through which the packet could not be forwarded. In ICMPv6 [RFC4443], the ICMP PTB message is a Packet Too Big Message with Code equal to (0). This message also indicates the MTU of the link through which the packet could not be forwarded.

2.2. Upper-layer Protocols

When an upper-layer protocol submits data to the underlying IP module, and the resulting IP packet's length is greater than the PMTU, IP fragmentation may be required. IP fragmentation divides a packet into fragments. Each fragment includes an IP header and a portion of the original packet.

[RFC0791] describes IPv4 fragmentation procedures. IPv4 packets whose DF-bit is set to one cannot be fragmented. IPv4 packets whose DF-bit is set to zero can be fragmented at the source node or by any downstream router. [RFC8200] describes IPv6 fragmentation procedures. IPv6 packets can be fragmented at the source node only.

IPv4 fragmentation differs slightly from IPv6 fragmentation. However, in both IP versions, the upper-layer header appears in the first fragment only. It does not appear in subsequent fragments.

Upper-layer protocols can operate in the following modes:

- o Do not rely on IP fragmentation.
- o Rely on IP source fragmentation only (i.e., fragmentation at the source node).
- o Rely on IP source fragmentation and downstream fragmentation (i.e., fragmentation at any node along the path).

Upper-layer protocols running over IPv4 can operate in all of the above-mentioned modes. Upper-layer protocols running over IPv6 can operate in the first and second modes only.

Upper-layer protocols that operate in the first two modes (above) require access to the PMTU estimate. In order to fulfil this requirement, they can

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures themselves.
- o Execute Packetization Layer PMTUD (PLPMTUD) [RFC4821] [I-D.fairhurst-tsvwg-datagram-plpmtud] procedures.

According to PLPMTUD procedures, the upper-layer protocol maintains a running PMTU estimate. It does so by sending probe packets of various sizes to its peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies on acknowledgement of received messages, as opposed to ICMP PTB messages concerning dropped messages. Therefore, PLPMTUD does not rely on the network's ability to deliver ICMP PTB messages to the source.

An upper-layer protocol that does not rely on IP fragmentation never causes the underlying IP module to emit

- o A fragmentable IP packet (i.e., an IPv4 packet with the DF-bit set to zero).
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

However, when the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

When an upper-layer protocol that relies on IP source fragmentation only submits data to the underlying IP module, and the resulting packet is larger than the PMTU estimate, the underlying IP module fragments the packet and emits the fragments. However, the upper-layer protocol never causes the underlying IP module to emit

- o A fragmentable IP packet.
- o A packet whose length is greater than the PMTU estimate.

When the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

An upper-layer protocol that relies on IP source fragmentation and downstream fragmentation can cause the underlying IP module to emit

- o A fragmentable IP packet.
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

A protocol that relies on IP source fragmentation and downstream fragmentation does not require access to the PMTU estimate. For these protocols, the underlying IP module:

- o Fragments all packets whose length exceeds the MTU of the first link along the path to the destination.
- o Sets the DF-bit to zero, so that downstream nodes can fragment the packet.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. IP Fragmentation Reduces Reliability

This section explains how IP fragmentation reduces the reliability of Internet communication.

4.1. Middle Box Failures

Many middle boxes require access to the transport-layer header. However, when a packet is divided into fragments, the transport-layer header appears in the first fragment only. It does not appear in

subsequent fragments. This omission can prevent middle boxes from delivering their intended services.

For example, assume that a router diverts selected packets from their normal path towards network appliances that support deep packet inspection and lawful intercept. The router selects packets for diversion based upon the following 5-tuple:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

IP fragmentation causes this selection algorithm to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In another example, a middle box remarks a packet's Differentiated Services Code Point [RFC2474] based upon the above-mentioned 5-tuple. IP fragmentation causes this process to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In all of the above-mentioned examples, the middle box cannot deliver its intended service without reassembling fragmented packets.

4.2. Partial Filtering

IP fragments cause problems for firewalls whose filter rules include decision making based on TCP and UDP ports. As the port information is not in the trailing fragments the firewall may elect to accept all trailing fragments, which may admit certain classes of attack, or may elect to block all trailing fragments, which may block otherwise legitimate traffic, or may elect to reassemble all fragmented packets, which may be inefficient and negatively affect performance.

4.3. Suboptimal Load Balancing

Many stateless load-balancers require access to the transport-layer header. Assume that a load-balancer distributes flows among parallel links. In order to optimize load balancing, the load-balancer sends every packet or packet fragment belonging to a flow through the same link.

In order to assign a packet or packet fragment to a link, the load-balancer executes an algorithm. If the packet or packet fragment contains a transport-layer header, the load balancing algorithm accepts the following 5-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

However, if the packet or packet fragment does not contain a transport-layer header, the load balancing algorithm accepts only the following 3-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.

Therefore, non-fragmented packets belonging to a flow can be assigned to one link while fragmented packets belonging to the same flow can be divided between that link and another. This can cause suboptimal load balancing.

4.4. Security Vulnerabilities

Security researchers have documented several attacks that rely on IP fragmentation. The following are examples:

- o Overlapping fragment attack [RFC1858][RFC3128] [RFC5722]
- o Resource exhaustion attacks (such as the Rose Attack)
- o Attacks based on predictable fragment identification values [RFC7739]
- o Attacks based on bugs in the implementation of the fragment reassembly algorithm
- o Evasion of Network Intrusion Detection Systems (NIDS) [Ptacek1998]

In the overlapping fragment attack, an attacker constructs a series of packet fragments. The first fragment contains an IP header, a transport-layer header, and some transport-layer payload. This fragment complies with local security policy and is allowed to pass through a stateless firewall. A second fragment, having a non-zero offset, overlaps with the first fragment. The second fragment also passes through the stateless firewall. When the packet is reassembled, the transport layer header from the first fragment is overwritten by data from the second fragment. The reassembled packet does not comply with local security policy. Had it traversed the firewall in one piece, the firewall would have rejected it.

A stateless firewall cannot protect against the overlapping fragment attack. However, destination nodes can protect against the overlapping fragment attack by implementing the reassembly procedures described in RFC 1858, RFC 3128 and RFC 8200. These reassembly procedures detect the overlap and discard the packet.

The fragment reassembly algorithm is a stateful procedure for an otherwise stateless protocol. As such, it can be exploited for resource exhaustion attacks. An attacker can construct a series of fragmented packets, with one fragment missing from each packet so that the reassembly process cannot complete. Thus, this attack causes resource exhaustion on the destination node, possibly denying reassembly services to other flows. This type of attack can be mitigated by flushing fragment reassembly buffers when necessary, at the expense of possibly dropping legitimate fragments.

An IP fragment contains an "Identification" field that, together with the IP Source Address and Destination Address of a packet, identifies fragments that correspond to the same original datagram, so that they can be reassembled together by the receiving host. Many implementations have employed predictable values for the Identification field, thus making it easy for an attacker to forge malicious IP fragments that would cause the reassembly procedure for legitimate packets to fail.

Over the years multiple IPv4 and IPv6 implementations have been found to have flaws in their implementation of the IP fragment reassembly algorithm, typically resulting in buffer overflows. These buffer overflows have been exploitable for denial of service and remote code execution attacks.

NIDS aims at identifying malicious activity by analyzing network traffic. Ambiguity in the possible result of the fragment reassembly process may allow an attacker to evade these systems. Many of these systems try to mitigate some of these evasion techniques by e.g.

computing all possible outcomes of the fragment reassembly process, at the expense of increased processing requirements.

4.5. Blackholing Due to ICMP Loss

As stated above, an upper-layer protocol requires access the PMTU estimate if it:

- o Does not rely on IP fragmentation.
- o Relies on IP source fragmentation only (i.e., fragmentation at the source node).

In order to satisfy this requirement, the upper-layer protocol can:

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures itself.
- o Execute PLPMTUD procedures.

PMTUD relies upon the network's ability to deliver ICMP PTB messages to the source node. Therefore, if an upper-layer protocol relies on PMTUD, it also relies on the network's ability to deliver ICMP PTB messages to the source node.

According to [RFC4890], ICMP PTB messages must not be filtered. However, ICMP PTB delivery is not reliable. It is subject to both transient and persistent loss.

Transient loss of ICMP PTB messages causes PMTUD to perform less efficiently, but does not cause it to fail completely. When the conditions contributing to transient loss abate, the network regains its ability to deliver ICMP PTB messages and PMTUD regains its ability to function. Section 4.5.1 of this document describes conditions that lead to transient loss of ICMP PTB messages.

However, persistent loss of ICMP PTB messages causes PMTUD to fail completely. Section 4.5.2 and Section 4.5.3 of this document describe conditions that lead to persistent loss of ICMP PTB messages.

The problem described in this section is specific to PMTUD. It does not occur when the upper-layer protocol obtains its PMTU estimate from PLPMTUD or any other source.

4.5.1. Transient Loss

The following factors can contribute to transient loss of ICMP PTB messages:

- o Network congestion.
- o Packet corruption.
- o Transient routing loops.
- o ICMP rate limiting.

The effect of rate limiting may be severe, as RFC 4443 recommends strict rate limiting of IPv6 traffic.

4.5.2. Incorrect Implementation of Security Policy

Incorrect implementation of security policy can cause persistent loss of ICMP PTB messages.

Assume that a Customer Premise Equipment (CPE) router implements the following zone-based security policy:

- o Allow any traffic to flow from the inside zone to the outside zone.
- o Do not allow any traffic to flow from the outside zone to the inside zone unless it is part of an existing flow (i.e., it was elicited by an outbound packet).

When a correct implementation of the above-mentioned security policy receives an ICMP PTB message, it examines the ICMP PTB payload in order to determine the original packet (i.e., the packet that elicited the ICMP PTB message) belonged to an existing flow. If the original packet belonged to an existing flow, the implementation allows the ICMP PTB to flow from the outside zone to the inside zone. If not, the implementation discards the ICMP PTB message.

When a incorrect implementation of the above-mentioned security policy receives an ICMP PTB message, it discards the packet because its source address is not associated with an existing flow.

The security policy described above is implemented incorrectly on many consumer CPE routers.

4.5.3. Persistent Loss Caused By Anycast

Anycast can cause persistent loss of ICMP PTB messages. Consider the example below:

A DNS client sends a request to an anycast address. The network routes that DNS request to the nearest instance of that anycast address (i.e., a DNS Server). The DNS server generates a response and sends it back to the DNS client. While the response does not exceed the DNS server's PMTU estimate, it does exceed the actual PMTU.

A downstream router drops the packet and sends an ICMP PTB message the packet's source (i.e., the anycast address). The network routes the ICMP PTB message to the anycast instance closest to the downstream router. Sadly, that anycast instance may not be the DNS server that originated the DNS response. It may be another DNS server with the same anycast address. The DNS server that originated the response may never receive the ICMP PTB message and may never update its PMTU estimate.

4.6. Blackholing Due To Filtering

In RFC 7872, researchers sampled Internet paths to determine whether they would convey packets that contain IPv6 extension headers. Sampled paths terminated at popular Internet sites (e.g., popular web, mail and DNS servers).

The study revealed that at least 28% of the sampled paths did not convey packets containing the IPv6 Fragment extension header. In most cases, fragments were dropped in the destination autonomous system. In other cases, the fragments were dropped in transit autonomous systems.

Another recent study [Huston] confirmed this finding. It reported that 37% of sampled endpoints used IPv6-capable DNS resolvers that were incapable of receiving a fragmented IPv6 response.

It is difficult to determine why network operators drop fragments. Possible causes follow:

- o Hardware inability to process fragmented packets.
- o Failure to change a vendor defaults.
- o Unintentional misconfiguration.

- o Intentional configuration (e.g., network operators consciously chooses to drop IPv6 fragments in order to address the issues raised in Section 4.1 through Section 4.5, above.)

5. Alternatives to IP Fragmentation

5.1. Transport Layer Solutions

The Transport Control Protocol (TCP) [RFC0793]) can be operated in a mode that does not require IP fragmentation.

Applications submit a stream of data to TCP. TCP divides that stream of data into segments, with no segment exceeding the TCP Maximum Segment Size (MSS). Each segment is encapsulated in a TCP header and submitted to the underlying IP module. The underlying IP module prepends an IP header and forwards the resulting packet.

If the TCP MSS is sufficiently small, the underlying IP module never produces a packet whose length is greater than the actual PMTU. Therefore, IP fragmentation is not required.

TCP offers the following mechanisms for MSS management:

- o Manual configuration
- o PMTUD
- o PLPMTUD

For IPv6 nodes, manual configuration is always applicable. If the MSS is manually configured to 1220 bytes and the packet does not contain extension headers, the IP layer will never produce a packet whose length is greater than the IPv6 minimum link MTU (1280 bytes). However, manual configuration prevents TCP from taking advantage of larger link MTU's.

RFC 8200 strongly recommends that IPv6 nodes implement PMTUD, in order to discover and take advantage of path MTUs greater than 1280 bytes. However, as mentioned in Section 2.1, PMTUD relies upon the network's ability to deliver ICMP PTB messages. Therefore, PMTUD is applicable only in environments where the risk of ICMP PTB loss is acceptable.

By contrast, PLPMTUD does not rely upon the network's ability to deliver ICMP PTB messages. However, in many loss-based TCP congestion control algorithms, the dropping of a packet may cause the TCP control algorithm to drop the congestion control window, or even re-start with the entire slow start process. For high capacity, long

round-trip time, large volume TCP streams, the deliberate probing with large packets and the consequent packet drop may impose too harsh a penalty on total TCP throughput for it to be a viable approach. [RFC4821] defines PLPMTUD procedures for TCP.

While TCP will never cause the underlying IP module to emit a packet that is larger than the PMTU estimate, it can cause the underlying IP module to emit a packet that is larger than the actual PMTU. If this occurs, the packet is dropped, the PMTU estimate is updated, the segment is divided into smaller segments and each smaller segment is submitted to the underlying IP module.

The Datagram Congestion Control Protocol (DCCP) [RFC4340] and the Stream Control Protocol (SCP) [RFC4960] also can be operated in a mode that does not require IP fragmentation. They both accept data from an application and divide that data into segments, with no segment exceeding a maximum size. Both DCCP and SCP offer manual configuration, PMTUD and PLPMTUD as mechanisms for managing that maximum size. [I-D.fairhurst-tsvwg-datagram-plpmtud] proposes PLPMTUD procedures for DCCP and SCP.

Currently, User Data Protocol (UDP) [RFC0768] lacks a fragmentation mechanism of its own and relies on IP fragmentation. However, [I-D.ietf-tsvwg-udp-options] proposes a fragmentation mechanism for UDP.

5.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. It also recognizes that UDP lacks a fragmentation mechanism of its own and relies on IP fragmentation. Therefore, [RFC8085] offers the following advice regarding applications the run over the UDP.

"An application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself to determine whether the path to a destination will support its desired message size without fragmentation."

RFC 8085 continues:

"Applications that do not follow the recommendation to do PMTU/PLPMTUD discovery SHOULD still avoid sending UDP datagrams that would result in IP packets that exceed the path MTU. Because the actual path MTU is unknown, such applications SHOULD fall back to sending

messages that are shorter than the default effective MTU for sending (EMTU_S in [RFC1122]). For IPv4, EMTU_S is the smaller of 576 bytes and the first-hop MTU. For IPv6, EMTU_S is 1280 bytes. The effective PMTU for a directly connected destination (with no routers on the path) is the configured interface MTU, which could be less than the maximum link payload size. Transmission of minimum-sized UDP datagrams is inefficient over paths that support a larger PMTU, which is a second reason to implement PMTU discovery."

RFC 8085 assumes that for IPv4, an EMTU_S of 576 is sufficiently small, even though the IPv4 minimum link MTU is 68 bytes.

This advice applies equally to application that run directly over IP.

6. Applications That Rely on IPv6 Fragmentation

The following applications rely on IPv6 fragmentation:

- o DNS [RFC1035]
- o OSPFv3 [RFC5340]
- o Packet-in-packet encapsulations

Each of these applications relies on IPv6 fragmentation to a varying degree. In some cases, that reliance is essential, and cannot be broken without fundamentally changing the protocol. In other cases, that reliance is incidental, and most implementations already take appropriate steps to avoid fragmentation.

This list is not comprehensive, and other protocols that rely on IPv6 fragmentation may exist. They are not specifically considered in the context of this document.

6.1. DNS

DNS relies on UDP for efficiency, and the consequence is the use of IP fragmentation for large responses, as permitted by the DNS EDNS(0) options in the query. It is possible to mitigate the issue of fragmentation-based packet loss by having queries use smaller EDNS(0) UDP buffer sizes, but then the operational issue of the partial level of support for DNS over TCP over IPv6 becomes a limiting factor of the efficacy of this approach in an IPv6 context [Damas].

Larger DNS responses can normally be avoided by aggressively pruning the Additional section of DNS responses. One scenario where such pruning is ineffective is in the use of DNSSEC, where large key sizes act to increase the response size to certain DNS queries. There is

no effective response to this situation within the DNS other than using smaller cryptographic keys and adoption of DNSSEC administrative practices that attempt to keep DNS response as short as possible.

6.2. OSPFv3

OSPFv3 implementations can emit messages large enough to cause IPv6 fragmentation. However, in keeping with the recommendations of RFC8200, and in order to optimize performance, most OSPFv3 implementations restrict their maximum message size to the IPv6 minimum link MTU.

6.3. Packet-in-Packet Encapsulations

In this document, packet-in-packet encapsulations include IP-in-IP [RFC2003], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086] and Generic Packet Tunneling in IPv6 [RFC2473]. [RFC4459] describes fragmentation issues associated with all of the above-mentioned encapsulations.

The fragmentation strategy described for GRE in [RFC7588] has been deployed for all of the above-mentioned encapsulations. This strategy does not rely on IPv6 fragmentation except in one corner case. (see Section 3.3.2.2 of RFC 7588 and Section 7.1 of RFC 2473). Section 3.3 of [RFC7676] further describes this corner case.

7. Recommendations

7.1. For Application Developers

Application developers SHOULD NOT develop applications that rely on IPv6 fragmentation.

Application-layer protocols then depend upon IPv6 fragmentation SHOULD be updated to break that dependency.

7.2. For Network Operators

As per RFC 4890, network operators MUST NOT filter ICMPv6 PTB messages unless they are known to be forged or otherwise illegitimate. As stated in Section 4.5, filtering ICMPv6 PTB packets causes PMTUD to fail. Operators MUST ensure proper PMTUD operation in their network, including making sure the network generates PTB packets when dropping packets too large compared to outgoing interface MTU.

Many upper-layer protocols rely on PMTUD.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

This document mitigates some of the security considerations associated with IP fragmentation by discouraging the use of IP fragmentation. It does not introduce any new security vulnerabilities, because it does not introduce any new alternatives to IP fragmentation. Instead, it recommends well-understood alternatives.

10. Acknowledgements

Thanks to Mikael Abrahamsson, Mike Heard, Tom Herbert, Tatuya Jinmei, Eric Nygren, and Joe Touch for their comments.

11. References

11.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.

- [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>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

11.2. Informative References

- [Damas] Damas, J. and G. Huston, "Measuring ATR", April 2018, <<http://www.potaroo.net/ispcol/2018-04/atr.html>>.
- [Huston] Huston, G., "IPv6, Large UDP Packets and the DNS (<http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html>)", August 2017.
- [I-D.fairhurst-tsvwg-datagram-plpmtud] Fairhurst, G., Jones, T., Tuexen, M., and I. Ruengeler, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-fairhurst-tsvwg-datagram-plpmtud-02 (work in progress), December 2017.

- [I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-02 (work in progress), January 2018.
- [Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <<http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, DOI 10.17487/RFC1858, October 1995, <<https://www.rfc-editor.org/info/rfc1858>>.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, DOI 10.17487/RFC3128, June 2001, <<https://www.rfc-editor.org/info/rfc3128>>.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <<https://www.rfc-editor.org/info/rfc4340>>.

- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, DOI 10.17487/RFC4459, April 2006, <<https://www.rfc-editor.org/info/rfc4459>>.
- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<https://www.rfc-editor.org/info/rfc4890>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <<https://www.rfc-editor.org/info/rfc5722>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <<https://www.rfc-editor.org/info/rfc5927>>.
- [RFC7588] Bonica, R., Pignataro, C., and J. Touch, "A Widely Deployed Solution to the Generic Routing Encapsulation (GRE) Fragmentation Problem", RFC 7588, DOI 10.17487/RFC7588, July 2015, <<https://www.rfc-editor.org/info/rfc7588>>.
- [RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <<https://www.rfc-editor.org/info/rfc7676>>.
- [RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.

[RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.

Appendix A. Contributors' Address

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Fred Baker
Unaffiliated
Santa Barbara, California 93117
USA

Email: FredBaker.IETF@gmail.com

Geoff Huston
APNIC
6 Cordelia St
Brisbane, 4101 QLD
Australia

Email: gih@apnic.net

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, California 94070
USA

Email: bob.hinden@gmail.com

Ole Troan
Cisco
Philip Pedersens vei 1
N-1366 Lysaker
Norway

Email: ot@cisco.com

Fernando Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires
Argentina

Email: fgont@si6networks.com

Internet Area WG
Internet-Draft
Intended status: Best Current Practice
Expires: January 24, 2019

R. Bonica
Juniper Networks
F. Baker
Unaffiliated
G. Huston
APNIC
R. Hinden
Check Point Software
O. Troan
Cisco
F. Gont
SI6 Networks
July 23, 2018

IP Fragmentation Considered Fragile
draft-bonica-intarea-frag-fragile-03

Abstract

This document provides an overview of IP fragmentation. It also explains how IP fragmentation reduces the reliability of Internet communication.

Finally, this document proposes alternatives to IP fragmentation and provides recommendations for application developers and network operators.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 24, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. IP Fragmentation	3
2.1. Links, Paths, MTU and PMTU	3
2.2. Upper-layer Protocols	5
3. Requirements Language	7
4. IP Fragmentation Reduces Reliability	7
4.1. Middle Box Failures	7
4.2. Partial Filtering	8
4.3. Telemetry and Monitoring and monitoring Failures	8
4.4. Suboptimal Load Balancing	9
4.5. Security Vulnerabilities	9
4.6. Blackholing Due to ICMP Loss	11
4.6.1. Transient Loss	12
4.6.2. Incorrect Implementation of Security Policy	12
4.6.3. Persistent Loss Caused By Anycast	13
4.7. Blackholing Due To Filtering	13
5. Alternatives to IP Fragmentation	14
5.1. Transport Layer Solutions	14
5.2. Application Layer Solutions	15
6. Applications That Rely on IPv6 Fragmentation	16
6.1. DNS	16
6.2. OSPFv3	17
6.3. Packet-in-Packet Encapsulations	17
7. Recommendations	17
7.1. For Application Developers	17
7.2. For Network Operators	18
8. IANA Considerations	18
9. Security Considerations	18
10. Acknowledgements	18
11. References	18
11.1. Normative References	18

11.2. Informative References	20
Appendix A. Contributors' Address	22
Authors' Addresses	22

1. Introduction

Operational experience [RFC7872] [Huston] reveals that IP fragmentation reduces the reliability of Internet communication. This document provides an overview of IP fragmentation. It also explains how IP fragmentation reduces the reliability of Internet communication.

Finally, this document proposes alternatives to IP fragmentation and provides recommendations for application developers and network operators.

2. IP Fragmentation

2.1. Links, Paths, MTU and PMTU

An Internet path connects a source node to a destination node. A path can contain links and intermediate systems. If a path contains more than one link, the links are connected in series and an intermediate system connects each link to the next. An intermediate system can be a router or a middle box.

Internet paths are dynamic. Assume that the path from one node to another contains a set of links and intermediate systems. If the network topology changes, that path can also change so that it includes a different set of links and intermediate systems.

Each link is constrained by the number of bytes that it can convey in a single IP packet. This constraint is called the link Maximum Transmission Unit (MTU). IPv4 [RFC0791] requires every link to have an MTU of 68 bytes or greater. IPv6 [RFC8200] requires every link to have an MTU of 1280 bytes or greater. These are called the IPv4 and IPv6 minimum link MTU's.

Each Internet path is constrained by the number of bytes that it can convey in a IP single packet. This constraint is called the Path MTU (PMTU). For any given path, the PMTU is equal to the smallest of its link MTU's. Because Internet paths are dynamic, PMTU is also dynamic.

For reasons described below, source nodes estimate the PMTU between themselves and destination nodes. A source node can produce extremely conservative PMTU estimates in which:

- o The estimate for each IPv4 path is equal to the IPv4 minimum link MTU.
- o The estimate for each IPv6 path is equal to the IPv6 minimum link MTU.

While these conservative estimates are guaranteed to be less than or equal to the actual PMTU, they are likely to be much less than the actual PMTU. This may adversely affect upper-layer protocol performance.

By executing Path MTU Discovery (PMTUD) [RFC1191] [RFC8201] procedures, a source node can maintain a less conservative, running estimate of the PMTU between itself and a destination node. According to these procedures, the source node produces an initial PMTU estimate. This initial estimate is equal to the MTU of the first link along the path to the destination node. It can be greater than the actual PMTU.

Having produced an initial PMTU estimate, the source node sends non-fragmentable IP packets to the destination node. If one of these packets is larger than the actual PMTU, a downstream router will not be able to forward the packet through the next link along the path. Therefore, the downstream router drops the packet and sends an Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443] Packet Too Big (PTB) message to the source node. The ICMP PTB message indicates the MTU of the link through which the packet could not be forwarded. The source node uses this information to refine its PMTU estimate.

PMTUD produces a running estimate of the PMTU between a source node and a destination node. Because PMTU is dynamic, at any given time, the PMTU estimate can differ from the actual PMTU. In order to detect PMTU increases, PMTUD occasionally resets the PMTU estimate to the MTU of the first link along path to the destination node. It then repeats the procedure described above.

PMTUD has the following characteristics:

- o It relies on the network's ability to deliver ICMP PTB messages to the source node.
- o It is susceptible to attack because ICMP messages are easily forged [RFC5927].

FOOTNOTE: According to RFC 0791, every IPv4 host must be capable of receiving a packet whose length is equal to 576 bytes. However, the

IPv4 minimum link MTU is not 576. Section 3.2 of RFC 0791 explicitly states that the IPv4 minimum link MTU is 68 bytes.

FOOTNOTE: In the paragraphs above, the term "non-fragmentable packet" is introduced. A non-fragmentable packet can be fragmented at its source. However, it cannot be fragmented by a downstream node. An IPv4 packet whose DF-bit is set to zero is fragmentable. An IPv4 packet whose DF-bit is set to one is non-fragmentable. All IPv6 packets are also non-fragmentable.

FOOTNOTE: In the paragraphs above, the term "ICMP PTB message" is introduced. The ICMP PTB message has two instantiations. In ICMPv4 [RFC0792], the ICMP PTB message is Destination Unreachable message with Code equal to (4) fragmentation needed and DF set. This message was augmented by [RFC1191] to indicate the MTU of the link through which the packet could not be forwarded. In ICMPv6 [RFC4443], the ICMP PTB message is a Packet Too Big Message with Code equal to (0). This message also indicates the MTU of the link through which the packet could not be forwarded.

2.2. Upper-layer Protocols

When an upper-layer protocol submits data to the underlying IP module, and the resulting IP packet's length is greater than the PMTU, IP fragmentation may be required. IP fragmentation divides a packet into fragments. Each fragment includes an IP header and a portion of the original packet.

[RFC0791] describes IPv4 fragmentation procedures. IPv4 packets whose DF-bit is set to one cannot be fragmented. IPv4 packets whose DF-bit is set to zero can be fragmented at the source node or by any downstream router. [RFC8200] describes IPv6 fragmentation procedures. IPv6 packets can be fragmented at the source node only.

IPv4 fragmentation differs slightly from IPv6 fragmentation. However, in both IP versions, the upper-layer header appears in the first fragment only. It does not appear in subsequent fragments.

Upper-layer protocols can operate in the following modes:

- o Do not rely on IP fragmentation.
- o Rely on IP source fragmentation only (i.e., fragmentation at the source node).
- o Rely on IP source fragmentation and downstream fragmentation (i.e., fragmentation at any node along the path).

Upper-layer protocols running over IPv4 can operate in all of the above-mentioned modes. Upper-layer protocols running over IPv6 can operate in the first and second modes only.

Upper-layer protocols that operate in the first two modes (above) require access to the PMTU estimate. In order to fulfil this requirement, they can

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures themselves.
- o Execute Packetization Layer PMTUD (PLPMTUD) [RFC4821] [I-D.fairhurst-tsvwg-datagram-plpmtud] procedures.

According to PLPMTUD procedures, the upper-layer protocol maintains a running PMTU estimate. It does so by sending probe packets of various sizes to its peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies on acknowledgement of received messages, as opposed to ICMP PTB messages concerning dropped messages. Therefore, PLPMTUD does not rely on the network's ability to deliver ICMP PTB messages to the source.

An upper-layer protocol that does not rely on IP fragmentation never causes the underlying IP module to emit

- o A fragmentable IP packet (i.e., an IPv4 packet with the DF-bit set to zero).
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

However, when the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

When an upper-layer protocol that relies on IP source fragmentation only submits data to the underlying IP module, and the resulting packet is larger than the PMTU estimate, the underlying IP module fragments the packet and emits the fragments. However, the upper-layer protocol never causes the underlying IP module to emit

- o A fragmentable IP packet.
- o A packet whose length is greater than the PMTU estimate.

When the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

An upper-layer protocol that relies on IP source fragmentation and downstream fragmentation can cause the underlying IP module to emit

- o A fragmentable IP packet.
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

A protocol that relies on IP source fragmentation and downstream fragmentation does not require access to the PMTU estimate. For these protocols, the underlying IP module:

- o Fragments all packets whose length exceeds the MTU of the first link along the path to the destination.
- o Sets the DF-bit to zero, so that downstream nodes can fragment the packet.

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. IP Fragmentation Reduces Reliability

This section explains how IP fragmentation reduces the reliability of Internet communication.

4.1. Middle Box Failures

Many middle boxes require access to the transport-layer header. However, when a packet is divided into fragments, the transport-layer header appears in the first fragment only. It does not appear in

subsequent fragments. This omission can prevent middle boxes from delivering their intended services.

For example, assume that a router diverts selected packets from their normal path towards network appliances that support deep packet inspection and lawful intercept. The router selects packets for diversion based upon the following 5-tuple:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

IP fragmentation causes this selection algorithm to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In another example, a middle box remarks a packet's Differentiated Services Code Point [RFC2474] based upon the above-mentioned 5-tuple. IP fragmentation causes this process to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In all of the above-mentioned examples, the middle box cannot deliver its intended service without reassembling fragmented packets.

4.2. Partial Filtering

IP fragments cause problems for firewalls whose filter rules include decision making based on TCP and UDP ports. As the port information is not in the trailing fragments the firewall may elect to accept all trailing fragments, which may admit certain classes of attack, or may elect to block all trailing fragments, which may block otherwise legitimate traffic, or may elect to reassemble all fragmented packets, which may be inefficient and negatively affect performance.

4.3. Telemetry and Monitoring and monitoring Failures

Stateless telemetry and monitoring strategies may require the transport-layer header to appear in every packet. However, when a packet is divided into fragments, the transport-layer header appears in the first fragment only. It does not appear in subsequent

fragments. This omission can prevent some stateless telemetry strategies from functioning correctly.

4.4. Suboptimal Load Balancing

Many stateless load-balancers require access to the transport-layer header. Assume that a load-balancer distributes flows among parallel links. In order to optimize load balancing, the load-balancer sends every packet or packet fragment belonging to a flow through the same link.

In order to assign a packet or packet fragment to a link, the load-balancer executes an algorithm. If the packet or packet fragment contains a transport-layer header, the load balancing algorithm accepts the following 5-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

However, if the packet or packet fragment does not contain a transport-layer header, the load balancing algorithm accepts only the following 3-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.

Therefore, non-fragmented packets belonging to a flow can be assigned to one link while fragmented packets belonging to the same flow can be divided between that link and another. This can cause suboptimal load balancing.

4.5. Security Vulnerabilities

Security researchers have documented several attacks that rely on IP fragmentation. The following are examples:

- o Overlapping fragment attack [RFC1858][RFC3128] [RFC5722]

- o Resource exhaustion attacks (such as the Rose Attack)
- o Attacks based on predictable fragment identification values [RFC7739]
- o Attacks based on bugs in the implementation of the fragment reassembly algorithm
- o Evasion of Network Intrusion Detection Systems (NIDS) [Ptacek1998]

In the overlapping fragment attack, an attacker constructs a series of packet fragments. The first fragment contains an IP header, a transport-layer header, and some transport-layer payload. This fragment complies with local security policy and is allowed to pass through a stateless firewall. A second fragment, having a non-zero offset, overlaps with the first fragment. The second fragment also passes through the stateless firewall. When the packet is reassembled, the transport layer header from the first fragment is overwritten by data from the second fragment. The reassembled packet does not comply with local security policy. Had it traversed the firewall in one piece, the firewall would have rejected it.

A stateless firewall cannot protect against the overlapping fragment attack. However, destination nodes can protect against the overlapping fragment attack by implementing the reassembly procedures described in RFC 1858, RFC 3128 and RFC 8200. These reassembly procedures detect the overlap and discard the packet.

The fragment reassembly algorithm is a stateful procedure for an otherwise stateless protocol. As such, it can be exploited for resource exhaustion attacks. An attacker can construct a series of fragmented packets, with one fragment missing from each packet so that the reassembly process cannot complete. Thus, this attack causes resource exhaustion on the destination node, possibly denying reassembly services to other flows. This type of attack can be mitigated by flushing fragment reassembly buffers when necessary, at the expense of possibly dropping legitimate fragments.

An IP fragment contains an "Identification" field that, together with the IP Source Address and Destination Address of a packet, identifies fragments that correspond to the same original datagram, so that they can be reassembled together by the receiving host. Many implementations have employed predictable values for the Identification field, thus making it easy for an attacker to forge malicious IP fragments that would cause the reassembly procedure for legitimate packets to fail.

Over the years multiple IPv4 and IPv6 implementations have been found to have flaws in their implementation of the IP fragment reassembly algorithm, typically resulting in buffer overflows. These buffer overflows have been exploitable for denial of service and remote code execution attacks.

NIDS aims at identifying malicious activity by analyzing network traffic. Ambiguity in the possible result of the fragment reassembly process may allow an attacker to evade these systems. Many of these systems try to mitigate some of these evasion techniques by e.g. Computing all possible outcomes of the fragment reassembly process, at the expense of increased processing requirements.

4.6. Blackholing Due to ICMP Loss

As stated above, an upper-layer protocol requires access the PMTU estimate if it:

- o Does not rely on IP fragmentation.
- o Relies on IP source fragmentation only (i.e., fragmentation at the source node).

In order to satisfy this requirement, the upper-layer protocol can:

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures itself.
- o Execute PLPMTUD procedures.

PMTUD relies upon the network's ability to deliver ICMP PTB messages to the source node. Therefore, if an upper-layer protocol relies on PMTUD, it also relies on the network's ability to deliver ICMP PTB messages to the source node.

According to [RFC4890], ICMP PTB messages must not be filtered. However, ICMP PTB delivery is not reliable. It is subject to both transient and persistent loss.

Transient loss of ICMP PTB messages causes PMTUD to perform less efficiently, but does not cause it to fail completely. When the conditions contributing to transient loss abate, the network regains its ability to deliver ICMP PTB messages and PMTUD regains its

ability to function. Section 4.6.1 of this document describes conditions that lead to transient loss of ICMP PTB messages.

However, persistent loss of ICMP PTB messages causes PMTUD to fail completely. Section 4.6.2 and Section 4.6.3 of this document describe conditions that lead to persistent loss of ICMP PTB messages.

The problem described in this section is specific to PMTUD. It does not occur when the upper-layer protocol obtains its PMTU estimate from PLPMTUD or any other source.

4.6.1. Transient Loss

The following factors can contribute to transient loss of ICMP PTB messages:

- o Network congestion.
- o Packet corruption.
- o Transient routing loops.
- o ICMP rate limiting.

The effect of rate limiting may be severe, as RFC 4443 recommends strict rate limiting of IPv6 traffic.

4.6.2. Incorrect Implementation of Security Policy

Incorrect implementation of security policy can cause persistent loss of ICMP PTB messages.

Assume that a Customer Premise Equipment (CPE) router implements the following zone-based security policy:

- o Allow any traffic to flow from the inside zone to the outside zone.
- o Do not allow any traffic to flow from the outside zone to the inside zone unless it is part of an existing flow (i.e., it was elicited by an outbound packet).

When a correct implementation of the above-mentioned security policy receives an ICMP PTB message, it examines the ICMP PTB payload in order to determine the original packet (i.e., the packet that elicited the ICMP PTB message) belonged to an existing flow. If the original packet belonged to an existing flow, the implementation

allows the ICMP PTB to flow from the outside zone to the inside zone. If not, the implementation discards the ICMP PTB message.

When a incorrect implementation of the above-mentioned security policy receives an ICMP PTB message, it discards the packet because its source address is not associated with an existing flow.

The security policy described above is implemented incorrectly on many consumer CPE routers.

4.6.3. Persistent Loss Caused By Anycast

Anycast can cause persistent loss of ICMP PTB messages. Consider the example below:

A DNS client sends a request to an anycast address. The network routes that DNS request to the nearest instance of that anycast address (i.e., a DNS Server). The DNS server generates a response and sends it back to the DNS client. While the response does not exceed the DNS server's PMTU estimate, it does exceed the actual PMTU.

A downstream router drops the packet and sends an ICMP PTB message the packet's source (i.e., the anycast address). The network routes the ICMP PTB message to the anycast instance closest to the downstream router. Sadly, that anycast instance may not be the DNS server that originated the DNS response. It may be another DNS server with the same anycast address. The DNS server that originated the response may never receive the ICMP PTB message and may never update its PMTU estimate.

4.7. Blackholing Due To Filtering

In RFC 7872, researchers sampled Internet paths to determine whether they would convey packets that contain IPv6 extension headers. Sampled paths terminated at popular Internet sites (e.g., popular web, mail and DNS servers).

The study revealed that at least 28% of the sampled paths did not convey packets containing the IPv6 Fragment extension header. In most cases, fragments were dropped in the destination autonomous system. In other cases, the fragments were dropped in transit autonomous systems.

Another recent study [Huston] confirmed this finding. It reported that 37% of sampled endpoints used IPv6-capable DNS resolvers that were incapable of receiving a fragmented IPv6 response.

It is difficult to determine why network operators drop fragments. Possible causes follow:

- o Hardware inability to process fragmented packets.
- o Failure to change a vendor defaults.
- o Unintentional misconfiguration.
- o Intentional configuration (e.g., network operators consciously chooses to drop IPv6 fragments in order to address the issues raised in Section 4.1 through Section 4.6, above.)

5. Alternatives to IP Fragmentation

5.1. Transport Layer Solutions

The Transport Control Protocol (TCP) [RFC0793]) can be operated in a mode that does not require IP fragmentation.

Applications submit a stream of data to TCP. TCP divides that stream of data into segments, with no segment exceeding the TCP Maximum Segment Size (MSS). Each segment is encapsulated in a TCP header and submitted to the underlying IP module. The underlying IP module prepends an IP header and forwards the resulting packet.

If the TCP MSS is sufficiently small, the underlying IP module never produces a packet whose length is greater than the actual PMTU. Therefore, IP fragmentation is not required.

TCP offers the following mechanisms for MSS management:

- o Manual configuration
- o PMTUD
- o PLPMTUD

For IPv6 nodes, manual configuration is always applicable. If the MSS is manually configured to 1220 bytes and the packet does not contain extension headers, the IP layer will never produce a packet whose length is greater than the IPv6 minimum link MTU (1280 bytes). However, manual configuration prevents TCP from taking advantage of larger link MTU's.

RFC 8200 strongly recommends that IPv6 nodes implement PMTUD, in order to discover and take advantage of path MTUs greater than 1280 bytes. However, as mentioned in Section 2.1, PMTUD relies upon the

network's ability to deliver ICMP PTB messages. Therefore, PMTUD is applicable only in environments where the risk of ICMP PTB loss is acceptable.

By contrast, PLPMTUD does not rely upon the network's ability to deliver ICMP PTB messages. However, in many loss-based TCP congestion control algorithms, the dropping of a packet may cause the TCP control algorithm to drop the congestion control window, or even re-start with the entire slow start process. For high capacity, long round-trip time, large volume TCP streams, the deliberate probing with large packets and the consequent packet drop may impose too harsh a penalty on total TCP throughput for it to be a viable approach. [RFC4821] defines PLPMTUD procedures for TCP.

While TCP will never cause the underlying IP module to emit a packet that is larger than the PMTU estimate, it can cause the underlying IP module to emit a packet that is larger than the actual PMTU. If this occurs, the packet is dropped, the PMTU estimate is updated, the segment is divided into smaller segments and each smaller segment is submitted to the underlying IP module.

The Datagram Congestion Control Protocol (DCCP) [RFC4340] and the Stream Control Protocol (SCP) [RFC4960] also can be operated in a mode that does not require IP fragmentation. They both accept data from an application and divide that data into segments, with no segment exceeding a maximum size. Both DCCP and SCP offer manual configuration, PMTUD and PLPMTUD as mechanisms for managing that maximum size. [I-D.fairhurst-tsvwg-datagram-plpmtud] proposes PLPMTUD procedures for DCCP and SCP.

Currently, User Data Protocol (UDP) [RFC0768] lacks a fragmentation mechanism of its own and relies on IP fragmentation. However, [I-D.ietf-tsvwg-udp-options] proposes a fragmentation mechanism for UDP.

5.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. It also recognizes that UDP lacks a fragmentation mechanism of its own and relies on IP fragmentation. Therefore, [RFC8085] offers the following advice regarding applications the run over the UDP.

"An application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself to determine whether the path to a

destination will support its desired message size without fragmentation."

RFC 8085 continues:

"Applications that do not follow the recommendation to do PMTU/PLPMTUD discovery SHOULD still avoid sending UDP datagrams that would result in IP packets that exceed the path MTU. Because the actual path MTU is unknown, such applications SHOULD fall back to sending messages that are shorter than the default effective MTU for sending (EMTU_S in [RFC1122]). For IPv4, EMTU_S is the smaller of 576 bytes and the first-hop MTU. For IPv6, EMTU_S is 1280 bytes. The effective PMTU for a directly connected destination (with no routers on the path) is the configured interface MTU, which could be less than the maximum link payload size. Transmission of minimum-sized UDP datagrams is inefficient over paths that support a larger PMTU, which is a second reason to implement PMTU discovery."

RFC 8085 assumes that for IPv4, an EMTU_S of 576 is sufficiently small, even though the IPv4 minimum link MTU is 68 bytes.

This advice applies equally to application that run directly over IP.

6. Applications That Rely on IPv6 Fragmentation

The following applications rely on IPv6 fragmentation:

- o DNS [RFC1035]
- o OSPFv3 [RFC5340]
- o Packet-in-packet encapsulations

Each of these applications relies on IPv6 fragmentation to a varying degree. In some cases, that reliance is essential, and cannot be broken without fundamentally changing the protocol. In other cases, that reliance is incidental, and most implementations already take appropriate steps to avoid fragmentation.

This list is not comprehensive, and other protocols that rely on IPv6 fragmentation may exist. They are not specifically considered in the context of this document.

6.1. DNS

DNS relies on UDP for efficiency, and the consequence is the use of IP fragmentation for large responses, as permitted by the DNS EDNS(0) options in the query. It is possible to mitigate the issue of

fragmentation-based packet loss by having queries use smaller EDNS(0) UDP buffer sizes, but then the operational issue of the partial level of support for DNS over TCP over IPv6 becomes a limiting factor of the efficacy of this approach in an IPv6 context [Damas].

Larger DNS responses can normally be avoided by aggressively pruning the Additional section of DNS responses. One scenario where such pruning is ineffective is in the use of DNSSEC, where large key sizes act to increase the response size to certain DNS queries. There is no effective response to this situation within the DNS other than using smaller cryptographic keys and adoption of DNSSEC administrative practices that attempt to keep DNS response as short as possible.

6.2. OSPFv3

OSPFv3 implementations can emit messages large enough to cause IPv6 fragmentation. However, in keeping with the recommendations of RFC8200, and in order to optimize performance, most OSPFv3 implementations restrict their maximum message size to the IPv6 minimum link MTU.

6.3. Packet-in-Packet Encapsulations

In this document, packet-in-packet encapsulations include IP-in-IP [RFC2003], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086] and Generic Packet Tunneling in IPv6 [RFC2473]. [RFC4459] describes fragmentation issues associated with all of the above-mentioned encapsulations.

The fragmentation strategy described for GRE in [RFC7588] has been deployed for all of the above-mentioned encapsulations. This strategy does not rely on IPv6 fragmentation except in one corner case. (see Section 3.3.2.2 of RFC 7588 and Section 7.1 of RFC 2473). Section 3.3 of [RFC7676] further describes this corner case.

7. Recommendations

7.1. For Application Developers

Application developers SHOULD NOT develop applications that rely on IPv6 fragmentation.

Application-layer protocols then depend upon IPv6 fragmentation SHOULD be updated to break that dependency.

7.2. For Network Operators

As per RFC 4890, network operators MUST NOT filter ICMPv6 PTB messages unless they are known to be forged or otherwise illegitimate. As stated in Section 4.6, filtering ICMPv6 PTB packets causes PMTUD to fail. Operators MUST ensure proper PMTUD operation in their network, including making sure the network generates PTB packets when dropping packets too large compared to outgoing interface MTU.

Many upper-layer protocols rely on PMTUD.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

This document mitigates some of the security considerations associated with IP fragmentation by discouraging the use of IP fragmentation. It does not introduce any new security vulnerabilities, because it does not introduce any new alternatives to IP fragmentation. Instead, it recommends well-understood alternatives.

10. Acknowledgements

Thanks to Mikael Abrahamsson, Lorenzo Colitti, Mike Heard, Tom Herbert, Tatuya Jinmei, Paolo Lucente, Eric Nygren, and Joe Touch for their comments.

11. References

11.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [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>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

11.2. Informative References

- [Damas] Damas, J. and G. Huston, "Measuring ATR", April 2018, <<http://www.potaroo.net/ispcol/2018-04/atr.html>>.
- [Huston] Huston, G., "IPv6, Large UDP Packets and the DNS (<http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html>)", August 2017.
- [I-D.fairhurst-tsvwg-datagram-plpmtud]
Fairhurst, G., Jones, T., Tuexen, M., and I. Ruengeler, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-fairhurst-tsvwg-datagram-plpmtud-02 (work in progress), December 2017.
- [I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-05 (work in progress), July 2018.
- [Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <<http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, DOI 10.17487/RFC1858, October 1995, <<https://www.rfc-editor.org/info/rfc1858>>.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.

- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, DOI 10.17487/RFC3128, June 2001, <<https://www.rfc-editor.org/info/rfc3128>>.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <<https://www.rfc-editor.org/info/rfc4340>>.
- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, DOI 10.17487/RFC4459, April 2006, <<https://www.rfc-editor.org/info/rfc4459>>.
- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<https://www.rfc-editor.org/info/rfc4890>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <<https://www.rfc-editor.org/info/rfc5722>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <<https://www.rfc-editor.org/info/rfc5927>>.
- [RFC7588] Bonica, R., Pignataro, C., and J. Touch, "A Widely Deployed Solution to the Generic Routing Encapsulation (GRE) Fragmentation Problem", RFC 7588, DOI 10.17487/RFC7588, July 2015, <<https://www.rfc-editor.org/info/rfc7588>>.

- [RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <<https://www.rfc-editor.org/info/rfc7676>>.
- [RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.

Appendix A. Contributors' Address

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Fred Baker
Unaffiliated
Santa Barbara, California 93117
USA

Email: FredBaker.IETF@gmail.com

Geoff Huston
APNIC
6 Cordelia St
Brisbane, 4101 QLD
Australia

Email: gih@apnic.net

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, California 94070
USA

Email: bob.hinden@gmail.com

Ole Troan
Cisco
Philip Pedersens vei 1
N-1366 Lysaker
Norway

Email: ot@cisco.com

Fernando Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires
Argentina

Email: fgont@si6networks.com

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

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 13, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	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 transitional 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 infrastructure. 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

- o "Possible": Means that the server software supports, in any way, the ability to record source ports for incoming connections.
- o "Feasible": Means that it there are no significant performance or storage implications for enabling the storage of source ports.
- o "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.
- o "Manageable": Means that tooling is, or has been, build or adapted to support the storage of source ports.
- o "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:

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

- o 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".
- o 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.
- o 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 timestamp 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 endpoint. Solutions to achieve this already exist for certain application layer protocols. The Forwarded HTTP Extension [RFC7239], for example, supports (as an optional feature) the transfer 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 address 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	OpenSSH	7.5	Yes	Yes	Yes

Table 2: Support for Logging Incoming Source Port

Author's Address

David O'Reilly
Ireland

Email: rfc@daveor.com

dprive
Internet-Draft
Intended status: Best Current Practice
Expires: January 3, 2019

S. Dickinson
Sinodun IT
B. Overeinder
NLnet Labs
R. van Rijswijk-Deij
SURFnet bv
A. Mankin
Salesforce
July 2, 2018

Recommendations for DNS Privacy Service Operators
draft-dickinson-dprive-bcp-op-00

Abstract

This document presents operational, policy and security considerations for DNS operators who choose to offer DNS Privacy services. With the recommendations, the operator can make deliberate decisions which services to provide, and how the decisions and alternatives impact the privacy of users.

This document also presents a framework to assist writers of DNS Privacy Policy and Practices Statements (analogous to DNS Security Extensions (DNSSEC) Policies and DNSSEC Practice Statements described in [RFC6841]).

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Scope	5
3. Privacy related documents	5
4. Terminology	6
5. Recommendations for DNS privacy services	6
5.1. On the wire between client and server	7
5.1.1. Transport recommendations	7
5.1.2. Authentication of DNS privacy services	8
5.1.3. Protocol recommendations	9
5.1.4. Availability	10
5.1.5. Service options	11
5.1.6. Limitations of using a pure TLS proxy	11
5.2. Data at rest on the server	12
5.2.1. Data handling	12
5.2.2. Data minimization of network traffic	13
5.2.3. IP address pseudonymization and anonymization methods	14
5.2.4. Pseudonymization, anonymization or discarding of other correlation data	14
5.2.5. Cache snooping	15
5.3. Data sent onwards from the server	15
5.3.1. Protocol recommendations	15
5.3.2. Client query obfuscation	16
5.3.3. Data sharing	17
6. DNS privacy policy and practice statement	17
6.1. Recommended contents of a DPPPS	18
6.2. Current policy and privacy statements	19
6.2.1. Quad9	19
6.2.2. Cloudflare	19
6.2.3. Google	20
6.2.4. OpenDNS	20
6.2.5. Comparison	20

6.3. Enforcement/accountability	20
7. IANA considerations	21
8. Security considerations	21
9. Acknowledgements	21
10. Contributors	21
11. Changelog	21
12. References	22
12.1. Normative References	22
12.2. Informative References	23
12.3. URIs	25
Appendix A. Documents	26
A.1. Potential increases in DNS privacy	26
A.2. Potential decreases in DNS privacy	27
A.3. Related operational documents	27
Appendix B. IP address techniques	28
B.1. Google Analytics non-prefix filtering	29
B.2. dnswasher	29
B.3. Prefix-preserving map	29
B.4. Cryptographic Prefix-Preserving Pseudonymisation	30
B.5. Top-hash Subtree-replicated Anonymisation	30
B.6. ipcipher	30
B.7. Bloom filters	31
Authors' Addresses	31

1. Introduction

[NOTE: This document is submitted to the IETF for initial review and for feedback on the best forum for future versions of this document. Initial considerations for DoH [I-D.ietf-doh-dns-over-https] are included here in anticipation of that draft progressing to be an RFC but further analysis is required.]

The Domain Name System (DNS) is at the core of the Internet; almost every activity on the Internet starts with a DNS query (and often several). However the DNS was not originally designed with strong security or privacy mechanisms. A number of developments have taken place in recent years which aim to increase the privacy of the DNS system and these are now seeing some deployment. This latest evolution of the DNS presents new challenges to operators and this document attempts to provide an overview of considerations for privacy focussed DNS services.

In recent years there has also been an increase in the availability of "open resolvers" [I-D.ietf-dnsop-terminology-bis] which users may prefer to use instead of the default network resolver because they offer a specific feature (e.g. good reachability, encrypted transport, strong privacy policy, filtering (or lack of), etc.). These open resolvers have tended to be at the forefront of adoption

of privacy related enhancements but it is anticipated that operators of other resolver services will follow.

Whilst protocols that encrypt DNS messages on the wire provide protection against certain attacks, the resolver operator still has (in principle) full visibility of the query data and transport identifiers for each user. Therefore, a trust relationship exists. The ability of the operator to provide a transparent, well documented, and secure privacy service will likely serve as a major differentiating factor for privacy conscious users if they make an active selection of which resolver to use.

It should also be noted that the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments has both advantages and disadvantages. For example the user has a clear expectation of which resolvers have visibility of their query data however this resolver/transport selection may provide an added mechanism to track them as they move across network environments. Commitments from operators to minimize such tracking are also likely to play a role in users selection of resolver.

More recently the global legislative landscape with regard to personal data collection, retention, and pseudonymization has seen significant activity with differing requirements active in different jurisdictions. For example the user of a service and the service itself may be in jurisdictions with conflicting legislation. It is an untested area that simply using a DNS resolution service constitutes consent from the user for the operator to process their query data. The impact of recent legislative changes on data pertaining to the users of both Internet Service Providers and DNS open resolvers is not fully understood at the time of writing.

This document has two main goals:

- o To provide operational and policy guidance related to DNS over encrypted transports and to outline recommendations for data handling for operators of DNS privacy services.
- o To introduce the DNS Privacy Policy and Practice Statement (DPPPS) and present a framework to assist writers of this document. A DPPPS is a document that an operator can publish outlining their operational practices and commitments with regard to privacy thereby providing a means for clients to evaluate the privacy properties of a given DNS privacy service. In particular, the framework identifies the elements that should be considered in formulating a DPPPS. This document does not, however, define a

particular Policy or Practice Statement, nor does it seek to provide legal advice or recommendations as to the contents.

Community insight [or judgment?] about operational practices can change quickly, and experience shows that a Best Current Practice (BCP) document about privacy and security is a point-in-time statement. Readers are advised to seek out any errata or updates that apply to this document.

2. Scope

"DNS Privacy Considerations" [RFC7626] describes the general privacy issues and threats associated with the use of the DNS by Internet users and much of the threat analysis here is lifted from that document and from [RFC6873]. However this document is limited in scope to best practice considerations for the provision of DNS privacy services by servers (recursive resolvers) to clients (stub resolvers or forwarders). Privacy considerations specifically from the perspective of an end user, or those for operators of authoritative nameservers are out of scope.

This document includes (but is not limited to) considerations in the following areas (taken from [RFC7626]):

1. Data "on the wire" between a client and a server
2. Data "at rest" on a server (e.g. in logs)
3. Data "sent onwards" from the server (either on the wire or shared with a third party)

Whilst the issues raised here are targeted at those operators who choose to offer a DNS privacy service, considerations for areas 2 and 3 could equally apply to operators who only offer DNS over unencrypted transports but who would like to align with privacy best practice.

3. Privacy related documents

There are various documents that describe protocol changes that have the potential to either increase or decrease the privacy of the DNS. Note this does not imply that some documents are good or bad, better or worse, just that (for example) some features may bring functional benefits at the price of a reduction in privacy and conversely some features increase privacy with an accompanying increase in complexity. A selection of the most relevant documents are listed in Appendix A for reference.

4. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Privacy terminology is as described in Section 3 of [RFC6973].

DNS terminology is as described in [I-D.ietf-dnsop-terminology-bis] with one modification: we use the definition of Privacy-enabling DNS server taken from [RFC8310]:

- o Privacy-enabling DNS server: A DNS server (most likely a full-service resolver) that implements DNS-over-TLS [RFC7858], and may optionally implement DNS-over-DTLS [RFC8094]. The server should also offer at least one of the credentials described in Section 8 and implement the (D)TLS profile described in Section 9.

TODO: Update the definition of Privacy-enabling DNS server in [I-D.ietf-dnsop-terminology-bis] to be complete and also include DoH, then reference that here.

- o DPPPS: DNS Privacy Policy and Practice Statement, see Section 6.
- o DNS privacy service: The service that is offered via a privacy-enabling DNS server and is documented either in an informal statement of policy and practice with regard to users privacy or a formal DPPPS.

5. Recommendations for DNS privacy services

We describe three classes of actions that operators of DNS privacy services can take:

- o Threat mitigation for well understood and documented privacy threats to the users of the service and in some cases to the operators of the service.
- o Optimization of privacy services from an operational or management perspective
- o Additional options that could further enhance the privacy and usability of the service

This document does not specify policy only best practice, however for DNS Privacy services to be considered compliant with these best practice guidelines they SHOULD implement (where appropriate) all:

- o Threat mitigations to be minimally compliant
- o Optimizations to be moderately compliant
- o Additional options to be maximally compliant

TODO: Some of the threats listed in the following sections are taken directly from Section 5 of RFC6973, some are just standalone descriptions, we need to go through all of them and see if we can use the RFC6973 threats where possible and make them consistent.

5.1. On the wire between client and server

In this section we consider both data on the wire and the service provided to the client.

5.1.1. Transport recommendations

Threats:

- o Surveillance: Passive surveillance of traffic on the wire
- o Intrusion: Active injection of spurious data or traffic

Mitigations:

A DNS privacy service can mitigate these threats by providing service over one or more of the following transports

- o DNS-over-TLS [RFC7858]
- o DoH [I-D.ietf-doh-dns-over-https]

Additional options:

- o A DNS privacy service can also be provided over DNS-over-DTLS [RFC8094], however note that this is an Experimental specification.

It is noted that DNS privacy service might be provided over IPSec, DNSCrypt or VPNs. However, use of these transports for DNS are not standardized and any discussion of best practice for providing such service is out of scope for this document.

5.1.2. Authentication of DNS privacy services

Threats:

- o Surveillance and Intrusion: Active attacks that can redirect traffic to rogue servers

Mitigations:

DNS privacy services should ensure clients can authenticate the server. Note that this, in effect, commits the DNS privacy service to a public identity users will trust.

When using DNS-over-TLS clients that select a 'Strict Privacy' usage profile [RFC8310] (to mitigate the threat of active attack on the client) require the ability to authenticate the DNS server. To enable this, DNS privacy services that offer DNS-over-TLS should provide credentials in the form of either X.509 certificates, SPKI pinsets or TLSA records.

When offering DoH [I-D.ietf-doh-dns-over-https], HTTPS requires authentication of the server as part of the protocol.

Optimizations:

DNS privacy services can also consider the following capabilities/options:

- o As recommended in [RFC8310] providing DANE TLSA records for the nameserver
 - * In particular, the service could provide TLSA records such that authenticating solely via the PKIX infrastructure can be avoided.
- o Implementing [I-D.ietf-tls-dnssec-chain-extension]
 - * This can decrease the latency of connection setup to the server and remove the need for the client to perform meta-queries to obtain and validate the DANE records.

5.1.2.1. Certificate management

Anecdotal evidence to date highlights the management of certificates as one of the more challenging aspects for operators of traditional DNS resolvers that choose to additionally provide a DNS privacy service as management of such credentials is new to those DNS operators.

It is noted that SPKI pinset management is described in [RFC7858] but that key pinning mechanisms in general have fallen out of favour operationally for various reasons.

Threats:

- o Invalid certificates, resulting in an unavailable service.
- o Mis-identification of a server by a client e.g. typos in URLs or authentication domain names

Mitigations:

It is recommended that operators:

- o Choose a short, memorable authentication name for their service
- o Automate the generation and publication of certificates
- o Monitor certificates to prevent accidental expiration of certificates

TODO: Could we provide references for certificate management best practice, for example Section 6.5 of RFC7525?

5.1.3. Protocol recommendations

5.1.3.1. DNS-over-TLS

Threats:

- o Known attacks on TLS (TODO: add a reference)
- o Traffic analysis (TODO: add a reference)
- o Potential for client tracking via transport identifiers
- o Blocking of well known ports (e.g. 853 for DNS-over-TLS)

Mitigations:

In the case of DNS-over-TLS, TLS profiles from Section 9 and the Countermeasures to DNS Traffic Analysis from section 11.1 of [RFC8310] provide strong mitigations. This includes but is not limited to:

- o Adhering to [RFC7525]

- o Implementing only (D)TLS 1.2 or later as specified in [RFC8310]
- o Implementing EDNS(0) Padding [RFC7830] using the guidelines in [I-D.ietf-dprive-padding-policy]
- o Clients should not be required to use TLS session resumption [RFC5077], Domain Name System (DNS) Cookies [RFC7873].
- o A DNS-over-TLS privacy service on both port 853 and 443. We note that this practice may require revision when DoH becomes more widely deployed, because of the potential use of the same ports for two incompatible types of service.

Optimizations:

- o Concurrent processing of pipelined queries, returning responses as soon as available, potentially out of order as specified in [RFC7766]. This is often called 'OOOR' - out-of-order responses. (Providing processing performance similar to HTTP multiplexing)
- o Management of TLS connections to optimize performance for clients using either
 - * [RFC7766] and EDNS(0) Keepalive [RFC7828] and/or
 - * DNS Stateful Operations [I-D.ietf-dnsop-session-signal]

Additional options that providers may consider:

- o Offer a .onion [RFC7686] service endpoint

5.1.3.2. DoH

TODO: Fill this in, a lot of overlap with DNS-over-TLS but we need to address DoH specific ones if possible.

Mitigations:

- o Clients should not be required to use HTTP Cookies [RFC6265].
- o Clients should not be required to include any headers beyond the absolute minimum to obtain service from a DoH server.

5.1.4. Availability

Threats:

- o A failed DNS privacy service could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service must be engineered for high availability. Particular care should be taken to protect DNS privacy services against denial-of-service attacks, as experience has shown that unavailability of DNS resolving because of attacks is a significant motivation for users to switch services.

TODO: Add reference to ongoing research on this topic.

5.1.5. Service options

Threats:

- o Unfairly disadvantaging users of the privacy service with respect to the services available. This could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service should deliver the same level of service offered on un-encrypted channels in terms of such options as filtering (or lack of), DNSSEC validation, etc.

5.1.6. Limitations of using a pure TLS proxy

Optimization:

Some operators may choose to implement DNS-over-TLS using a TLS proxy (e.g. nginx [1], haproxy [2] or stunnel [3]) in front of a DNS nameserver because of proven robustness and capacity when handling large numbers of client connections, load balancing capabilities and good tooling. Currently, however, because such proxies typically have no specific handling of DNS as a protocol over TLS or DTLS using them can restrict traffic management at the proxy layer and at the DNS server. For example, all traffic received by a nameserver behind such a proxy will appear to originate from the proxy and DNS techniques such as ACLs, RRL or DNS64 will be hard or impossible to implement in the nameserver.

Operators may choose to use a DNS aware proxy such as dnsdist.

5.2. Data at rest on the server

5.2.1. Data handling

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

The following are common activities for DNS service operators and in all cases should be minimized or completely avoided if possible for DNS privacy services. If data is retained it should be encrypted and either aggregated, pseudonymized or anonymized whenever possible. In general the principle of data minimization described in [RFC6973] should be applied.

- o Transient data (e.g. that is used for real time monitoring and threat analysis which might be held only memory) should be retained for the shortest possible period deemed operationally feasible.
- o The retention period of DNS traffic logs should be only those required to sustain operation of the service and, to the extent that such exists, meet regulatory requirements.
- o DNS privacy services should not track users except for the particular purpose of detecting and remedying technically malicious (e.g. DoS) or anomalous use of the service.
- o Data access should be minimized to only those personal who require access to perform operational duties.

5.2.2. Data minimization of network traffic

Data minimization refers to collecting, using, disclosing, and storing the minimal data necessary to perform a task, and this can be achieved by removing or obfuscating privacy-sensitive information in network traffic logs. This is typically personal data, or data that can be used to link a record to an individual, but may also include revealing other confidential information, for example on the structure of an internal corporate network.

The problem of effectively ensuring that DNS traffic logs contain no or minimal privacy-sensitive information is not one that currently has a generally agreed solution or any Standards to inform this discussion. This section presents an overview of current techniques to simply provide reference on the current status of this work.

Research into data minimization techniques (and particularly IP address pseudonymization/anonymization) was sparked in the late 1990s/early 2000s, partly driven by the desire to share significant corpuses of traffic captures for research purposes. Several techniques reflecting different requirements in this area and different performance/resource tradeoffs emerged over the course of the decade. Developments over the last decade have been both a blessing and a curse; the large increase in size between an IPv4 and an IPv6 address, for example, renders some techniques impractical, but also makes available a much larger amount of input entropy, the better to resist brute force re-identification attacks that have grown in practicality over the period.

Techniques employed may be broadly categorized as either anonymization or pseudonymization. The following discussion uses the definitions from [RFC6973] Section 3, with additional observations from van Dijkhuizen et al. [4]

- o Anonymization. To enable anonymity of an individual, there must exist a set of individuals that appear to have the same attribute(s) as the individual. To the attacker or the observer, these individuals must appear indistinguishable from each other.
- o Pseudonymization. The true identity is deterministically replaced with an alternate identity (a pseudonym). When the pseudonymization schema is known, the process can be reversed, so the original identity becomes known again.

In practice there is a fine line between the two; for example, how to categorize a deterministic algorithm for data minimization of IP addresses that produces a group of pseudonyms for a single given address.

5.2.3. IP address pseudonymization and anonymization methods

As [RFC7626] makes clear, the big privacy risk in DNS is connecting DNS queries to an individual and the major vector for this in DNS traffic is the client IP address.

There is active discussion in the space of effective pseudonymization of IP addresses in DNS traffic logs, however there seems to be no single solution that is widely recognized as suitable for all or most use cases. There are also as yet no standards for this that are unencumbered by patents. This following table presents a high level comparison of various techniques employed or under development today and classifies them according to categorization of technique and other properties. The list of techniques includes the main techniques in current use, but does not claim to be comprehensive. Appendix B provides a more detailed survey of these techniques and definitions for the categories and properties listed below.

Figure showing comparison of IP address techniques (SVG) [5]

The choice of which method to use for a particular application will depend on the requirements of that application and consideration of the threat analysis of the particular situation.

For example, a common goal is that distributed packet captures must be in an existing data format such as PCAP [pcap] or C-DNS [I-D.ietf-dnsop-dns-capture-format] that can be used as input to existing analysis tools. In that case, use of a Format-preserving technique is essential. This, though, is not cost-free - several authors (e.g. Brenker & Arnes [6]) have observed that, as the entropy in a IPv4 address is limited, given a de-identified log from a target, if an attacker is capable of ensuring packets are captured by the target and the attacker can send forged traffic with arbitrary source and destination addresses to that target, any format-preserving pseudonymization is vulnerable to an attack along the lines of a cryptographic chosen plaintext attack.

5.2.4. Pseudonymization, anonymization or discarding of other correlation data

Threats:

- o IP TTL/Hoplimit can be used to fingerprint client OS
- o Tracking of TCP sessions
- o Tracking of TLS sessions and session resumption mechanisms

- o Resolvers might receive client identifiers e.g. MAC addresses in EDNS(0) options - some CPE devices are known to add them.

- o HTTP headers

Mitigations:

- o Data minimization or discarding of such correlation data

TODO: More analysis here.

5.2.5. Cache snooping

Threats:

- o Profiling of client queries by malicious third parties

Mitigations:

TODO: Describe techniques to defend against cache snooping

5.3. Data sent onwards from the server

In this section we consider both data sent on the wire in upstream queries and data shared with third parties.

5.3.1. Protocol recommendations

Threats:

- o Transmission of identifying data upstream.

Mitigations:

As specified in [RFC8310] for DNS-over-TLS but applicable to any DNS Privacy services the server should:

- o Implement QNAME minimization [RFC7816]
- o Honour a SOURCE PREFIX-LENGTH set to 0 in a query containing the EDNS(0) Client Subnet (ECS) option and not send an ECS option in upstream queries.

Optimizations:

- o The server should either
 - * not use the ECS option in upstream queries at all, or

- * offer alternative services, one that sends ECS and one that does not.

If operators do offer a service that sends the ECS options upstream they should use the shortest prefix that is operationally feasible (NOTE: the authors believe they will be able to add a reference for advice here soon) and ideally use a policy of whitelisting upstream servers to send ECS to in order to minimize data leakage. Operators should make clear in any policy statement what prefix length they actually send and the specific policy used.

Additional options:

- o Aggressive Use of DNSSEC-Validated Cache [RFC8198] to reduce the number of queries to authoritative servers to increase privacy.
- o Run a copy of the root zone on loopback [RFC7706] to avoid making queries to the root servers that might leak information.

5.3.2. Client query obfuscation

Additional options:

Since queries from recursive resolvers to authoritative servers are performed using cleartext (at the time of writing), resolver services need to consider the extent to which they may be directly leaking information about their client community via these upstream queries and what they can do to mitigate this further. Note, that even when all the relevant techniques described above are employed there may still be attacks possible, e.g. [Pitfalls-of-DNS-Encryption]. For example, a resolver with a very small community of users risks exposing data in this way and OUGHT obfuscate this traffic by mixing it with 'generated' traffic to make client characterization harder. The resolver could also employ aggressive pre-fetch techniques as a further measure to counter traffic analysis.

At the time of writing there are no standardized or widely recognized techniques to preform such obfuscation or bulk pre-fetches.

Another technique that particularly small operators may consider is forwarding local traffic to a larger resolver (with a privacy policy that aligns with their own practices) over an encrypted protocol so that the upstream queries are obfuscated among those of the large resolver.

5.3.3. Data sharing

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

Operators should not provide identifiable data to third-parties without explicit consent from clients (we take the stance here that simply using the resolution service itself does not constitute consent).

Even when consent is granted operators should employ data minimization techniques such as those described in Section 5.2.1 if data is shared with third-parties.

Operators should consider including specific guidelines for the collection of aggregated and/or anonymized data for research purposes, within or outside of their own organization.

TODO: More on data for research vs operations... how to still motivate operators to share anonymized data?

TODO: Guidelines for when consent is granted?

TODO: Applies to server data handling too.. could operators offer alternatives services one that implies consent for data processing, one that doesn't?

6. DNS privacy policy and practice statement

6.1. Recommended contents of a DPPPS

1 Policy

1.1 Recommendations. This section should explain, with reference to section Section 5 of this document which recommendations the DNS privacy service employs.

1.2 Data handling. This section should explain, with reference to section Section 5.2 of this document the policy for gathering and disseminating information collected by the DNS privacy service.

1.2.1 Specify clearly what data (including whether it is aggregated, pseudonymized or anonymized) is:

1.2.1.1 Collected and retained by the operator (and for how long)

1.2.1.2 Shared with partners

1.2.1.3 Shared, sold or rented to third-parties

1.2.2 Specify any exceptions to the above, for example technically malicious or anomalous behaviour

1.2.3 Declare any partners, third-party affiliations or sources of funding

1.2.4 Whether user DNS data is correlated or combined with any other personal information held by the operator

2 Practice. This section should explain the current operational practices of the service.

2.1 Specify any temporary or permanent deviations from the policy for operational reasons

2.2 With reference to section Section 5.1 provide specific details of which capabilities are provided on which address and ports

2.3 With reference to section Section 5.3 provide specific details of which capabilities are employed for upstream traffic from the server for

2.4 Specify the authentication name to be used (if any) and if TLSA records are published (including options used in the TLSA records)

2.5 Specify the SPKI pinsets to be used (if any) and policy for rolling keys

2.6 Provide a contact email address for the service

6.2. Current policy and privacy statements

NOTE: An analysis of these statements will clearly only provide a snapshot at the time of writing. It is included in this version of the draft to provide a basis for the assessment of the contents of the DPPPS and is expected to be removed or substantially re-worked in a future version.

6.2.1. Quad9

UDP/TCP and TLS (port 853) service provided on two addresses:

- o 'Secure': 9.9.9.9, 149.112.112.112, 2620:fe::fe, 2620:fe::9
- o 'Unsecured': 9.9.9.10, 149.112.112.10, 2620:fe::10

Policy:

- o <<https://www.quad9.net/policy/>>
- o <<https://www.quad9.net/privacy/>>
- o <<https://www.quad9.net/faq/>>

6.2.2. Cloudflare

UDP/TCP and TLS (port 853) service provided on 1.1.1.1, 1.0.0.1, 2606:4700:4700::1111 and 2606:4700:4700::1001.

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/privacy-policy/>>

DoH provided on: <<https://cloudflare-dns.com/dns-query>>

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/firefox/>>

Tor endpoint: <<https://dns4torpnlfs2ifuz2s2yf3fc7rdmsbhm6rw75euj35pac6ap25zgqad.onion>>.

6.2.3. Google

UDP/TCP service provided on 8.8.8.8, 8.8.4.4, 2001:4860:4860::8888 and 2001:4860:4860::8844.

Policy: <<https://developers.google.com/speed/public-dns/privacy>>

6.2.4. OpenDNS

UDP/TCP service provided on 208.67.222.222 and 208.67.220.220 (no IPv6).

We could find no specific privacy policy for the DNS resolution, only a general one from Cisco that seems focussed on websites.

Policy: <<https://www.cisco.com/c/en/us/about/legal/privacy-full.html>>

6.2.5. Comparison

The following tables provides a high-level comparison of the policy and practice statements above and also some observations of practice measured at dnsprivacy.org [7]. The data is not exhaustive and has not been reviewed or confirmed by the operators.

A question mark indicates no clear statement or data could be located on the issue. A dash indicates the category is not applicable to the service.

Table showing comparison of operators policies [8]

Table showing comparison of operators practices [9]

NOTE: Review and correction of any inaccuracies in the table would be much appreciated.

6.3. Enforcement/accountability

Transparency reports may help with building user trust that operators adhere to their policies and practices.

Independent monitoring should be performed where possible of:

- o ECS, QNAME minimization, EDNS(0) padding, etc.
- o Filtering
- o Uptime

7. IANA considerations

None

8. Security considerations

TODO: e.g. New issues for DoS defence, server admin policies

9. Acknowledgements

Many thanks to Amelia Andersdotter for a very thorough review of the first draft of this document. Thanks also to John Todd for discussions on this topic, and to Stephane Bortzmeyer for review.

Sara Dickinson thanks the Open Technology Fund for a grant to support the work on this document.

10. Contributors

The below individuals contributed significantly to the document:

John Dickinson
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Jim Hague
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

11. Changelog

draft-dickinson-dprive-bcp-op-00

Name change to add dprive. Differences to draft-dickinson-bcp-op-00:

- o Reworked the Terminology, Introduction and Scope
- o Added Document section
- o Reworked the Recommendations section to describe threat mitigations, optimizations and other options. Split the

recommendations up into 3 subsections: on the wire, at rest and upstream

- o Added much more information on data handling and IP address pseudonymization and anonymization
- o Added more details and comparison of some existing policy/privacy policies
- o Applied virtually all of Amelia Andersdotter's suggested changes.

draft-dickinson-bcp-op-00

- o Initial commit

12. References

12.1. Normative References

- [I-D.ietf-dnsop-terminology-bis]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-10 (work in progress), April 2018.
- [I-D.ietf-doh-dns-over-https]
Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", draft-ietf-doh-dns-over-https-12 (work in progress), June 2018.
- [I-D.ietf-dprive-padding-policy]
Mayrhofer, A., "Padding Policy for EDNS(0)", draft-ietf-dprive-padding-policy-05 (work in progress), April 2018.
- [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>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <<https://www.rfc-editor.org/info/rfc5077>>.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <<https://www.rfc-editor.org/info/rfc6265>>.

- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830, DOI 10.17487/RFC7830, May 2016, <<https://www.rfc-editor.org/info/rfc7830>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <<https://www.rfc-editor.org/info/rfc7873>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.

12.2. Informative References

- [I-D.ietf-dnsop-dns-capture-format]
Dickinson, J., Hague, J., Dickinson, S., Manderson, T.,
and J. Bond, "C-DNS: A DNS Packet Capture Format", draft-
ietf-dnsop-dns-capture-format-07 (work in progress), May
2018.
- [I-D.ietf-dnsop-dns-tcp-requirements]
Kristoff, J. and D. Wessels, "DNS Transport over TCP -
Operational Requirements", draft-ietf-dnsop-dns-tcp-
requirements-02 (work in progress), May 2018.
- [I-D.ietf-dnsop-session-signal]
Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S.,
Lemon, T., and T. Pusateri, "DNS Stateful Operations",
draft-ietf-dnsop-session-signal-10 (work in progress),
June 2018.
- [I-D.ietf-tls-dnssec-chain-extension]
Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE
Record and DNSSEC Authentication Chain Extension for TLS",
draft-ietf-tls-dnssec-chain-extension-07 (work in
progress), March 2018.
- [pcap] tcpdump.org, "PCAP", 2016, <<http://www.tcpdump.org/>>.
- [Pitfalls-of-DNS-Encryption]
Shulman, H., "Pretty Bad Privacy: Pitfalls of DNS
Encryption", 2014, <[https://www.ietf.org/mail-archive/web/
dns-privacy/current/pdfWqAIUmEl47.pdf](https://www.ietf.org/mail-archive/web/dns-privacy/current/pdfWqAIUmEl47.pdf)>.
- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization
Support", RFC 6235, DOI 10.17487/RFC6235, May 2011,
<<https://www.rfc-editor.org/info/rfc6235>>.
- [RFC6841] Ljunggren, F., Eklund Lowinder, AM., and T. Okubo, "A
Framework for DNSSEC Policies and DNSSEC Practice
Statements", RFC 6841, DOI 10.17487/RFC6841, January 2013,
<<https://www.rfc-editor.org/info/rfc6841>>.
- [RFC6873] Salgueiro, G., Gurbani, V., and A. Roach, "Format for the
Session Initiation Protocol (SIP) Common Log Format
(CLF)", RFC 6873, DOI 10.17487/RFC6873, February 2013,
<<https://www.rfc-editor.org/info/rfc6873>>.
- [RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use
Domain Name", RFC 7686, DOI 10.17487/RFC7686, October
2015, <<https://www.rfc-editor.org/info/rfc7686>>.

- [RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <<https://www.rfc-editor.org/info/rfc7706>>.
- [RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <<https://www.rfc-editor.org/info/rfc7766>>.
- [RFC7828] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The edns-tcp-keepalive EDNS0 Option", RFC 7828, DOI 10.17487/RFC7828, April 2016, <<https://www.rfc-editor.org/info/rfc7828>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <<https://www.rfc-editor.org/info/rfc8094>>.
- [RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.

12.3. URIs

- [1] <https://nginx.org/>
- [2] <https://www.haproxy.org/>
- [3] <https://kb.isc.org/article/AA-01386/0/DNS-over-TLS.html>
- [4] <https://doi.org/10.1145/3182660>
- [5] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/ip_techniques_table.svg
- [6] <https://pdfs.semanticscholar.org/7b34/12c951cebe71cd2cddac5fda164fb2138a44.pdf>
- [7] <https://dnsprivacy.org/jenkins/job/dnsprivacy-monitoring/>

- [8] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/policy_table.svg
- [9] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/practice_table.svg
- [10] <https://support.google.com/analytics/answer/2763052?hl=en>
- [11] <https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-geo-impact-test/>
- [12] <https://github.com/edmonds/pdns/blob/master/pdns/dnswasher.cc>
- [13] <http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html>
- [14] <http://an.kaist.ac.kr/~sbmoon/paper/intl-journal/2004-cn-anon.pdf>
- [15] <https://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/>
- [16] http://mharvan.net/talks/noms-ip_anon.pdf
- [17] <https://medium.com/@bert.hubert/on-ip-address-encryption-security-analysis-with-respect-for-privacy-dabel201b476>
- [18] <https://github.com/PowerDNS/ipcipher>
- [19] <https://github.com/veorq/ipcrypt>
- [20] <https://www.ietf.org/mail-archive/web/cfrg/current/msg09494.html>
- [21] <https://tncl8.geant.org/core/presentation/127>

Appendix A. Documents

This section provides an overview of some DNS privacy related documents, however, this is neither an exhaustive list nor a definitive statement on the characteristic of the document.

A.1. Potential increases in DNS privacy

These documents are limited in scope to communications between stub clients and recursive resolvers:

- o 'Specification for DNS over Transport Layer Security (TLS)' [RFC7858], referred to here as 'DNS-over-TLS'.

- o 'DNS over Datagram Transport Layer Security (DTLS)' [RFC8094], referred to here as 'DNS-over-DTLS'. Note that this document has the Category of Experimental.
- o 'DNS Queries over HTTPS (DoH)' [I-D.ietf-doh-dns-over-https] referred to here as DoH.
- o 'Usage Profiles for DNS over TLS and DNS over DTLS' [RFC8310]
- o 'The EDNS(0) Padding Option' [RFC7830] and 'Padding Policy for EDNS(0)' [I-D.ietf-dprive-padding-policy]

These documents apply to recursive to authoritative DNS but are relevant when considering the operation of a recursive server:

- o 'DNS Query Name minimization to Improve Privacy' [RFC7816] referred to here as 'QNAME minimization'

A.2. Potential decreases in DNS privacy

These documents relate to functionality that could provide increased tracking of user activity as a side effect:

- o 'Client Subnet in DNS Queries' [RFC7871]
- o 'Domain Name System (DNS) Cookies' [RFC7873])
- o 'Transport Layer Security (TLS) Session Resumption without Server-Side State' [RFC5077] referred to here as simply TLS session resumption.
- o 'A DNS Packet Capture Format' [I-D.ietf-dnsop-dns-capture-format]
- o Passive DNS [I-D.ietf-dnsop-terminology-bis]

Note that depending on the specifics of the implementation [I-D.ietf-doh-dns-over-https] may also provide increased tracking.

A.3. Related operational documents

- o 'DNS Transport over TCP - Implementation Requirements' [RFC7766]
- o 'Operational requirements for DNS-over-TCP' [I-D.ietf-dnsop-dns-tcp-requirements]
- o 'The edns-tcp-keepalive EDNS0 Option' [RFC7828]
- o 'DNS Stateful Operations' [I-D.ietf-dnsop-session-signal]

Appendix B. IP address techniques

Data minimization methods may be categorized by the processing used and the properties of their outputs. The following builds on the categorization employed in [RFC6235]:

- o Format-preserving. Normally when encrypting, the original data length and patterns in the data should be hidden from an attacker. Some applications of de-identification, such as network capture de-identification, require that the de-identified data is of the same form as the original data, to allow the data to be parsed in the same way as the original.
- o Prefix preservation. Values such as IP addresses and MAC addresses contain prefix information that can be valuable in analysis, e.g. manufacturer ID in MAC addresses, subnet in IP addresses. Prefix preservation ensures that prefixes are de-identified consistently; e.g. if two IP addresses are from the same subnet, a prefix preserving de-identification will ensure that their de-identified counterparts will also share a subnet. Prefix preservation may be fixed - the extent of the prefix to be preserved must be identified in advance - or general.
- o Replacement. A one-to-one replacement of a field to a new value of the same type, for example using a regular expression. Filtering. Removing (and thus truncating) or replacing data in a field. Field data can be overwritten, often with zeros, either partially (grey marking) or completely (black marking).
- o Generalization. Data is replaced by more general data with reduced specificity. One example would be to replace all TCP/UDP port numbers with one of two fixed values indicating whether the original port was ephemeral (≥ 1024) or non-ephemeral (> 1024). Another example, precision degradation, reduces the accuracy of e.g. a numeric value or a timestamp.
- o Enumeration. With data from a well-ordered set, replace the first data item data using a random initial value and then allocate ordered values for subsequent data items. When used with timestamp data, this preserves ordering but loses precision and distance.
- o Reordering/shuffling. Preserving the original data, but rearranging its order, often in a random manner.
- o Random substitution. As replacement, but using randomly generated replacement values.

- o Cryptographic permutation. Using a permutation function, such as a hash function or cryptographic block cipher, to generate a replacement de-identified value.

B.1. Google Analytics non-prefix filtering

Since May 2010, Google Analytics has provided a facility [10] that allows website owners to request that all their users IP addresses are anonymized within Google Analytics processing. This very basic anonymization simply sets to zero the least significant 8 bits of IPv4 addresses, and the least significant 80 bits of IPv6 addresses. The level of anonymization this produces is perhaps questionable. There are some analysis results [11] which suggest that the impact of this on reducing the accuracy of determining the user's location from their IP address is less than might be hoped; the average discrepancy in identification of the user city for UK users is no more than 17%.

Anonymization: Format-preserving, Filtering (grey marking).

B.2. dnswasher

Since 2006, PowerDNS have included a de-identification tool dnswasher [12] with their PowerDNS product. This is a PCAP filter that performs a one-to-one mapping of end user IP addresses with an anonymized address. A table of user IP addresses and their de-identified counterparts is kept; the first IPv4 user addresses is translated to 0.0.0.1, the second to 0.0.0.2 and so on. The de-identified address therefore depends on the order that addresses arrive in the input, and running over a large amount of data the address translation tables can grow to a significant size.

Anonymization: Format-preserving, Enumeration.

B.3. Prefix-preserving map

Used in TCPdpriv [13], this algorithm stores a set of original and anonymised IP address pairs. When a new IP address arrives, it is compared with previous addresses to determine the longest prefix match. The new address is anonymized by using the same prefix, with the remainder of the address anonymized with a random value. The use of a random value means that the TCPdpriv is not deterministic; different anonymized values will be generated on each run. The need to store previous addresses means that TCPdpriv has significant and unbounded memory requirements, and because of the need to allocated anonymized addresses sequentially cannot be used in parallel processing.

Anonymization: Format-preserving, prefix preservation (general).

B.4. Cryptographic Prefix-Preserving Pseudonymisation

Cryptographic prefix-preserving pseudonymisation was originally proposed as an improvement to the prefix-preserving map implemented in TCPdpriv, described in Xu et al. [14] and implemented in the Crypto-PAN tool [15]. Crypto-PAN is now frequently used as an acronym for the algorithm. Initially it was described for IPv4 addresses only; extension for IPv6 addresses was proposed in Harvan & Schoenwaelder [16] and implemented in snmpdump. This uses a cryptographic algorithm rather than a random value, and thus pseudonymity is determined uniquely by the encryption key, and is deterministic. It requires a separate AES encryption for each output bit, so has a non-trivial calculation overhead. This can be mitigated to some extent (for IPv4, at least) by pre-calculating results for some number of prefix bits.

Pseudonymization: Format-preserving, prefix preservation (general).

B.5. Top-hash Subtree-replicated Anonymisation

Proposed in Ramaswamy & Wolf, Top-hash Subtree-replicated Anonymisation (TSA) originated in response to the requirement for faster processing than Crypto-PAN. It used hashing for the most significant byte of an IPv4 address, and a pre-calculated binary tree structure for the remainder of the address. To save memory space, replication is used within the tree structure, reducing the size of the pre-calculated structures to a few Mb for IPv4 addresses. Address pseudonymization is done via hash and table lookup, and so requires minimal computation. However, due to the much increased address space for IPv6, TSA is not memory efficient for IPv6.

Pseudonymization: Format-preserving, prefix preservation (general).

B.6. ipcipher

A recently-released proposal from PowerDNS [17], ipcipher [18] is a simple pseudonymization technique for IPv4 and IPv6 addresses. IPv6 addresses are encrypted directly with AES-128 using a key (which may be derived from a passphrase). IPv4 addresses are similarly encrypted, but using a recently proposed encryption ipcrypt [19] suitable for 32bit block lengths. However, the author of ipcrypt has since indicated [20] that it has low security, and further analysis has revealed it is vulnerable to attack.

Pseudonymization: Format-preserving, cryptographic permutation.

B.7. Bloom filters

van Rijswijk-Deij et al. [21] have recently described work using Bloom filters to categorize query traffic and record the traffic as the state of multiple filters. The goal of this work is to allow operators to identify so-called Indicators of Compromise (IOCs) originating from specific subnets without storing information about, or be able to monitor the DNS queries of an individual user. By using a Bloom filter, it is possible to determine with a high probability if, for example, a particular query was made, but the set of queries made cannot be recovered from the filter. Similarly, by mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to determine if a particular user performed a particular query. Large numbers of queries can be tracked in a memory-efficient way. As filter status is stored, this approach cannot be used to regenerate traffic, and so cannot be used with tools used to process live traffic.

Anonymized: Generalization.

Authors' Addresses

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Email: sara@sinodun.com

Benno J. Overeinder
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: benno@nlnetLabs.nl

Roland M. van Rijswijk-Deij
SURFnet bv
PO Box 19035
Utrecht 3501 DA Utrecht
The Netherlands

Email: roland.vanrijswijk@surfnet.nl

Allison Mankin
Salesforce

Email: allison.mankin@gmail.com

dprive
Internet-Draft
Intended status: Best Current Practice
Expires: January 17, 2019

S. Dickinson
Sinodun IT
B. Overeinder
NLnet Labs
R. van Rijswijk-Deij
SURFnet bv
A. Mankin
Salesforce
July 16, 2018

Recommendations for DNS Privacy Service Operators
draft-dickinson-dprive-bcp-op-01

Abstract

This document presents operational, policy and security considerations for DNS operators who choose to offer DNS Privacy services. With the recommendations, the operator can make deliberate decisions which services to provide, and how the decisions and alternatives impact the privacy of users.

This document also presents a framework to assist writers of DNS Privacy Policy and Practices Statements (analogous to DNS Security Extensions (DNSSEC) Policies and DNSSEC Practice Statements described in [RFC6841]).

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 17, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Scope	5
3. Privacy related documents	5
4. Terminology	6
5. Recommendations for DNS privacy services	6
5.1. On the wire between client and server	7
5.1.1. Transport recommendations	7
5.1.2. Authentication of DNS privacy services	8
5.1.3. Protocol recommendations	9
5.1.4. Availability	10
5.1.5. Service options	11
5.1.6. Limitations of using a pure TLS proxy	11
5.2. Data at rest on the server	12
5.2.1. Data handling	12
5.2.2. Data minimization of network traffic	13
5.2.3. IP address pseudonymization and anonymization methods	14
5.2.4. Pseudonymization, anonymization or discarding of other correlation data	14
5.2.5. Cache snooping	15
5.3. Data sent onwards from the server	15
5.3.1. Protocol recommendations	15
5.3.2. Client query obfuscation	16
5.3.3. Data sharing	17
6. DNS privacy policy and practice statement	17
6.1. Recommended contents of a DPPPS	18
6.2. Current policy and privacy statements	19
6.2.1. Quad9	19
6.2.2. Cloudflare	19
6.2.3. Google	20
6.2.4. OpenDNS	20
6.2.5. Comparison	20

6.3. Enforcement/accountability	20
7. IANA considerations	21
8. Security considerations	21
9. Acknowledgements	21
10. Contributors	21
11. Changelog	21
12. References	22
12.1. Normative References	22
12.2. Informative References	23
12.3. URIs	25
Appendix A. Documents	26
A.1. Potential increases in DNS privacy	26
A.2. Potential decreases in DNS privacy	27
A.3. Related operational documents	27
Appendix B. IP address techniques	28
B.1. Google Analytics non-prefix filtering	29
B.2. dnswasher	29
B.3. Prefix-preserving map	29
B.4. Cryptographic Prefix-Preserving Pseudonymisation	30
B.5. Top-hash Subtree-replicated Anonymisation	30
B.6. ipcipher	30
B.7. Bloom filters	31
Authors' Addresses	31

1. Introduction

[NOTE: This document is submitted to the IETF for initial review and for feedback on the best forum for future versions of this document. Initial considerations for DoH [I-D.ietf-doh-dns-over-https] are included here in anticipation of that draft progressing to be an RFC but further analysis is required.]

The Domain Name System (DNS) is at the core of the Internet; almost every activity on the Internet starts with a DNS query (and often several). However the DNS was not originally designed with strong security or privacy mechanisms. A number of developments have taken place in recent years which aim to increase the privacy of the DNS system and these are now seeing some deployment. This latest evolution of the DNS presents new challenges to operators and this document attempts to provide an overview of considerations for privacy focussed DNS services.

In recent years there has also been an increase in the availability of "open resolvers" [I-D.ietf-dnsop-terminology-bis] which users may prefer to use instead of the default network resolver because they offer a specific feature (e.g. good reachability, encrypted transport, strong privacy policy, filtering (or lack of), etc.). These open resolvers have tended to be at the forefront of adoption

of privacy related enhancements but it is anticipated that operators of other resolver services will follow.

Whilst protocols that encrypt DNS messages on the wire provide protection against certain attacks, the resolver operator still has (in principle) full visibility of the query data and transport identifiers for each user. Therefore, a trust relationship exists. The ability of the operator to provide a transparent, well documented, and secure privacy service will likely serve as a major differentiating factor for privacy conscious users if they make an active selection of which resolver to use.

It should also be noted that the choice of a user to configure a single resolver (or a fixed set of resolvers) and an encrypted transport to use in all network environments has both advantages and disadvantages. For example the user has a clear expectation of which resolvers have visibility of their query data however this resolver/transport selection may provide an added mechanism to track them as they move across network environments. Commitments from operators to minimize such tracking are also likely to play a role in users selection of resolver.

More recently the global legislative landscape with regard to personal data collection, retention, and pseudonymization has seen significant activity with differing requirements active in different jurisdictions. For example the user of a service and the service itself may be in jurisdictions with conflicting legislation. It is an untested area that simply using a DNS resolution service constitutes consent from the user for the operator to process their query data. The impact of recent legislative changes on data pertaining to the users of both Internet Service Providers and DNS open resolvers is not fully understood at the time of writing.

This document has two main goals:

- o To provide operational and policy guidance related to DNS over encrypted transports and to outline recommendations for data handling for operators of DNS privacy services.
- o To introduce the DNS Privacy Policy and Practice Statement (DPPPS) and present a framework to assist writers of this document. A DPPPS is a document that an operator can publish outlining their operational practices and commitments with regard to privacy thereby providing a means for clients to evaluate the privacy properties of a given DNS privacy service. In particular, the framework identifies the elements that should be considered in formulating a DPPPS. This document does not, however, define a

particular Policy or Practice Statement, nor does it seek to provide legal advice or recommendations as to the contents.

Community insight [or judgment?] about operational practices can change quickly, and experience shows that a Best Current Practice (BCP) document about privacy and security is a point-in-time statement. Readers are advised to seek out any errata or updates that apply to this document.

2. Scope

"DNS Privacy Considerations" [I-D.bortzmeyer-dprive-rfc7626-bis] describes the general privacy issues and threats associated with the use of the DNS by Internet users and much of the threat analysis here is lifted from that document and from [RFC6873]. However this document is limited in scope to best practice considerations for the provision of DNS privacy services by servers (recursive resolvers) to clients (stub resolvers or forwarders). Privacy considerations specifically from the perspective of an end user, or those for operators of authoritative nameservers are out of scope.

This document includes (but is not limited to) considerations in the following areas (taken from [I-D.bortzmeyer-dprive-rfc7626-bis]):

1. Data "on the wire" between a client and a server
2. Data "at rest" on a server (e.g. in logs)
3. Data "sent onwards" from the server (either on the wire or shared with a third party)

Whilst the issues raised here are targeted at those operators who choose to offer a DNS privacy service, considerations for areas 2 and 3 could equally apply to operators who only offer DNS over unencrypted transports but who would like to align with privacy best practice.

3. Privacy related documents

There are various documents that describe protocol changes that have the potential to either increase or decrease the privacy of the DNS. Note this does not imply that some documents are good or bad, better or worse, just that (for example) some features may bring functional benefits at the price of a reduction in privacy and conversely some features increase privacy with an accompanying increase in complexity. A selection of the most relevant documents are listed in Appendix A for reference.

4. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Privacy terminology is as described in Section 3 of [RFC6973].

DNS terminology is as described in [I-D.ietf-dnsop-terminology-bis] with one modification: we use the definition of Privacy-enabling DNS server taken from [RFC8310]:

- o Privacy-enabling DNS server: A DNS server (most likely a full-service resolver) that implements DNS-over-TLS [RFC7858], and may optionally implement DNS-over-DTLS [RFC8094]. The server should also offer at least one of the credentials described in Section 8 and implement the (D)TLS profile described in Section 9.

TODO: Update the definition of Privacy-enabling DNS server in [I-D.ietf-dnsop-terminology-bis] to be complete and also include DoH, then reference that here.

- o DPPPS: DNS Privacy Policy and Practice Statement, see Section 6.
- o DNS privacy service: The service that is offered via a privacy-enabling DNS server and is documented either in an informal statement of policy and practice with regard to users privacy or a formal DPPPS.

5. Recommendations for DNS privacy services

We describe three classes of actions that operators of DNS privacy services can take:

- o Threat mitigation for well understood and documented privacy threats to the users of the service and in some cases to the operators of the service.
- o Optimization of privacy services from an operational or management perspective
- o Additional options that could further enhance the privacy and usability of the service

This document does not specify policy only best practice, however for DNS Privacy services to be considered compliant with these best practice guidelines they SHOULD implement (where appropriate) all:

- o Threat mitigations to be minimally compliant
- o Optimizations to be moderately compliant
- o Additional options to be maximally compliant

TODO: Some of the threats listed in the following sections are taken directly from Section 5 of RFC6973, some are just standalone descriptions, we need to go through all of them and see if we can use the RFC6973 threats where possible and make them consistent.

5.1. On the wire between client and server

In this section we consider both data on the wire and the service provided to the client.

5.1.1. Transport recommendations

Threats:

- o Surveillance: Passive surveillance of traffic on the wire
- o Intrusion: Active injection of spurious data or traffic

Mitigations:

A DNS privacy service can mitigate these threats by providing service over one or more of the following transports

- o DNS-over-TLS [RFC7858]
- o DoH [I-D.ietf-doh-dns-over-https]

Additional options:

- o A DNS privacy service can also be provided over DNS-over-DTLS [RFC8094], however note that this is an Experimental specification.

It is noted that DNS privacy service might be provided over IPSec, DNSCrypt or VPNs. However, use of these transports for DNS are not standardized and any discussion of best practice for providing such service is out of scope for this document.

5.1.2. Authentication of DNS privacy services

Threats:

- o Surveillance and Intrusion: Active attacks that can redirect traffic to rogue servers

Mitigations:

DNS privacy services should ensure clients can authenticate the server. Note that this, in effect, commits the DNS privacy service to a public identity users will trust.

When using DNS-over-TLS clients that select a 'Strict Privacy' usage profile [RFC8310] (to mitigate the threat of active attack on the client) require the ability to authenticate the DNS server. To enable this, DNS privacy services that offer DNS-over-TLS should provide credentials in the form of either X.509 certificates, SPKI pinsets or TLSA records.

When offering DoH [I-D.ietf-doh-dns-over-https], HTTPS requires authentication of the server as part of the protocol.

Optimizations:

DNS privacy services can also consider the following capabilities/options:

- o As recommended in [RFC8310] providing DANE TLSA records for the nameserver
 - * In particular, the service could provide TLSA records such that authenticating solely via the PKIX infrastructure can be avoided.
- o Implementing [I-D.ietf-tls-dnssec-chain-extension]
 - * This can decrease the latency of connection setup to the server and remove the need for the client to perform meta-queries to obtain and validate the DANE records.

5.1.2.1. Certificate management

Anecdotal evidence to date highlights the management of certificates as one of the more challenging aspects for operators of traditional DNS resolvers that choose to additionally provide a DNS privacy service as management of such credentials is new to those DNS operators.

It is noted that SPKI pinset management is described in [RFC7858] but that key pinning mechanisms in general have fallen out of favour operationally for various reasons.

Threats:

- o Invalid certificates, resulting in an unavailable service.
- o Mis-identification of a server by a client e.g. typos in URLs or authentication domain names

Mitigations:

It is recommended that operators:

- o Choose a short, memorable authentication name for their service
- o Automate the generation and publication of certificates
- o Monitor certificates to prevent accidental expiration of certificates

TODO: Could we provide references for certificate management best practice, for example Section 6.5 of RFC7525?

5.1.3. Protocol recommendations

5.1.3.1. DNS-over-TLS

Threats:

- o Known attacks on TLS (TODO: add a reference)
- o Traffic analysis (TODO: add a reference)
- o Potential for client tracking via transport identifiers
- o Blocking of well known ports (e.g. 853 for DNS-over-TLS)

Mitigations:

In the case of DNS-over-TLS, TLS profiles from Section 9 and the Countermeasures to DNS Traffic Analysis from section 11.1 of [RFC8310] provide strong mitigations. This includes but is not limited to:

- o Adhering to [RFC7525]

- o Implementing only (D)TLS 1.2 or later as specified in [RFC8310]
- o Implementing EDNS(0) Padding [RFC7830] using the guidelines in [I-D.ietf-dprive-padding-policy]
- o Clients should not be required to use TLS session resumption [RFC5077], Domain Name System (DNS) Cookies [RFC7873].
- o A DNS-over-TLS privacy service on both port 853 and 443. We note that this practice may require revision when DoH becomes more widely deployed, because of the potential use of the same ports for two incompatible types of service.

Optimizations:

- o Concurrent processing of pipelined queries, returning responses as soon as available, potentially out of order as specified in [RFC7766]. This is often called 'OOOR' - out-of-order responses. (Providing processing performance similar to HTTP multiplexing)
- o Management of TLS connections to optimize performance for clients using either
 - * [RFC7766] and EDNS(0) Keepalive [RFC7828] and/or
 - * DNS Stateful Operations [I-D.ietf-dnsop-session-signal]

Additional options that providers may consider:

- o Offer a .onion [RFC7686] service endpoint

5.1.3.2. DoH

TODO: Fill this in, a lot of overlap with DNS-over-TLS but we need to address DoH specific ones if possible.

Mitigations:

- o Clients should not be required to use HTTP Cookies [RFC6265].
- o Clients should not be required to include any headers beyond the absolute minimum to obtain service from a DoH server.

5.1.4. Availability

Threats:

- o A failed DNS privacy service could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service must be engineered for high availability. Particular care should be taken to protect DNS privacy services against denial-of-service attacks, as experience has shown that unavailability of DNS resolving because of attacks is a significant motivation for users to switch services.

TODO: Add reference to ongoing research on this topic.

5.1.5. Service options

Threats:

- o Unfairly disadvantaging users of the privacy service with respect to the services available. This could force the user to switch providers, fallback to cleartext or accept no DNS service for the outage.

Mitigations:

A DNS privacy service should deliver the same level of service offered on un-encrypted channels in terms of such options as filtering (or lack of), DNSSEC validation, etc.

5.1.6. Limitations of using a pure TLS proxy

Optimization:

Some operators may choose to implement DNS-over-TLS using a TLS proxy (e.g. nginx [1], haproxy [2] or stunnel [3]) in front of a DNS nameserver because of proven robustness and capacity when handling large numbers of client connections, load balancing capabilities and good tooling. Currently, however, because such proxies typically have no specific handling of DNS as a protocol over TLS or DTLS using them can restrict traffic management at the proxy layer and at the DNS server. For example, all traffic received by a nameserver behind such a proxy will appear to originate from the proxy and DNS techniques such as ACLs, RRL or DNS64 will be hard or impossible to implement in the nameserver.

Operators may choose to use a DNS aware proxy such as dnsdist.

5.2. Data at rest on the server

5.2.1. Data handling

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

The following are common activities for DNS service operators and in all cases should be minimized or completely avoided if possible for DNS privacy services. If data is retained it should be encrypted and either aggregated, pseudonymized or anonymized whenever possible. In general the principle of data minimization described in [RFC6973] should be applied.

- o Transient data (e.g. that is used for real time monitoring and threat analysis which might be held only memory) should be retained for the shortest possible period deemed operationally feasible.
- o The retention period of DNS traffic logs should be only those required to sustain operation of the service and, to the extent that such exists, meet regulatory requirements.
- o DNS privacy services should not track users except for the particular purpose of detecting and remedying technically malicious (e.g. DoS) or anomalous use of the service.
- o Data access should be minimized to only those personal who require access to perform operational duties.

5.2.2. Data minimization of network traffic

Data minimization refers to collecting, using, disclosing, and storing the minimal data necessary to perform a task, and this can be achieved by removing or obfuscating privacy-sensitive information in network traffic logs. This is typically personal data, or data that can be used to link a record to an individual, but may also include revealing other confidential information, for example on the structure of an internal corporate network.

The problem of effectively ensuring that DNS traffic logs contain no or minimal privacy-sensitive information is not one that currently has a generally agreed solution or any Standards to inform this discussion. This section presents an overview of current techniques to simply provide reference on the current status of this work.

Research into data minimization techniques (and particularly IP address pseudonymization/anonymization) was sparked in the late 1990s/early 2000s, partly driven by the desire to share significant corpuses of traffic captures for research purposes. Several techniques reflecting different requirements in this area and different performance/resource tradeoffs emerged over the course of the decade. Developments over the last decade have been both a blessing and a curse; the large increase in size between an IPv4 and an IPv6 address, for example, renders some techniques impractical, but also makes available a much larger amount of input entropy, the better to resist brute force re-identification attacks that have grown in practicality over the period.

Techniques employed may be broadly categorized as either anonymization or pseudonymization. The following discussion uses the definitions from [RFC6973] Section 3, with additional observations from van Dijkhuizen et al. [4]

- o Anonymization. To enable anonymity of an individual, there must exist a set of individuals that appear to have the same attribute(s) as the individual. To the attacker or the observer, these individuals must appear indistinguishable from each other.
- o Pseudonymization. The true identity is deterministically replaced with an alternate identity (a pseudonym). When the pseudonymization schema is known, the process can be reversed, so the original identity becomes known again.

In practice there is a fine line between the two; for example, how to categorize a deterministic algorithm for data minimization of IP addresses that produces a group of pseudonyms for a single given address.

5.2.3. IP address pseudonymization and anonymization methods

As [I-D.bortzmeyer-dprive-rfc7626-bis] makes clear, the big privacy risk in DNS is connecting DNS queries to an individual and the major vector for this in DNS traffic is the client IP address.

There is active discussion in the space of effective pseudonymization of IP addresses in DNS traffic logs, however there seems to be no single solution that is widely recognized as suitable for all or most use cases. There are also as yet no standards for this that are unencumbered by patents. This following table presents a high level comparison of various techniques employed or under development today and classifies them according to categorization of technique and other properties. The list of techniques includes the main techniques in current use, but does not claim to be comprehensive. Appendix B provides a more detailed survey of these techniques and definitions for the categories and properties listed below.

Figure showing comparison of IP address techniques (SVG) [5]

The choice of which method to use for a particular application will depend on the requirements of that application and consideration of the threat analysis of the particular situation.

For example, a common goal is that distributed packet captures must be in an existing data format such as PCAP [pcap] or C-DNS [I-D.ietf-dnsop-dns-capture-format] that can be used as input to existing analysis tools. In that case, use of a Format-preserving technique is essential. This, though, is not cost-free - several authors (e.g. Brenker & Arnes [6]) have observed that, as the entropy in a IPv4 address is limited, given a de-identified log from a target, if an attacker is capable of ensuring packets are captured by the target and the attacker can send forged traffic with arbitrary source and destination addresses to that target, any format-preserving pseudonymization is vulnerable to an attack along the lines of a cryptographic chosen plaintext attack.

5.2.4. Pseudonymization, anonymization or discarding of other correlation data

Threats:

- o IP TTL/Hoplimit can be used to fingerprint client OS
- o Tracking of TCP sessions
- o Tracking of TLS sessions and session resumption mechanisms

- o Resolvers *might* receive client identifiers e.g. MAC addresses in EDNS(0) options - some CPE devices are known to add them.

- o HTTP headers

Mitigations:

- o Data minimization or discarding of such correlation data

TODO: More analysis here.

5.2.5. Cache snooping

Threats:

- o Profiling of client queries by malicious third parties

Mitigations:

TODO: Describe techniques to defend against cache snooping

5.3. Data sent onwards from the server

In this section we consider both data sent on the wire in upstream queries and data shared with third parties.

5.3.1. Protocol recommendations

Threats:

- o Transmission of identifying data upstream.

Mitigations:

As specified in [RFC8310] for DNS-over-TLS but applicable to any DNS Privacy services the server should:

- o Implement QNAME minimization [RFC7816]
- o Honour a SOURCE PREFIX-LENGTH set to 0 in a query containing the EDNS(0) Client Subnet (ECS) option and not send an ECS option in upstream queries.

Optimizations:

- o The server should either
 - * not use the ECS option in upstream queries at all, or

- * offer alternative services, one that sends ECS and one that does not.

If operators do offer a service that sends the ECS options upstream they should use the shortest prefix that is operationally feasible (NOTE: the authors believe they will be able to add a reference for advice here soon) and ideally use a policy of whitelisting upstream servers to send ECS to in order to minimize data leakage. Operators should make clear in any policy statement what prefix length they actually send and the specific policy used.

Additional options:

- o Aggressive Use of DNSSEC-Validated Cache [RFC8198] to reduce the number of queries to authoritative servers to increase privacy.
- o Run a copy of the root zone on loopback [RFC7706] to avoid making queries to the root servers that might leak information.

5.3.2. Client query obfuscation

Additional options:

Since queries from recursive resolvers to authoritative servers are performed using cleartext (at the time of writing), resolver services need to consider the extent to which they may be directly leaking information about their client community via these upstream queries and what they can do to mitigate this further. Note, that even when all the relevant techniques described above are employed there may still be attacks possible, e.g. [Pitfalls-of-DNS-Encryption]. For example, a resolver with a very small community of users risks exposing data in this way and OUGHT obfuscate this traffic by mixing it with 'generated' traffic to make client characterization harder. The resolver could also employ aggressive pre-fetch techniques as a further measure to counter traffic analysis.

At the time of writing there are no standardized or widely recognized techniques to preform such obfuscation or bulk pre-fetches.

Another technique that particularly small operators may consider is forwarding local traffic to a larger resolver (with a privacy policy that aligns with their own practices) over an encrypted protocol so that the upstream queries are obfuscated among those of the large resolver.

5.3.3. Data sharing

Threats:

- o Surveillance
- o Stored data compromise
- o Correlation
- o Identification
- o Secondary use
- o Disclosure
- o Contravention of legal requirements not to process user data?

Mitigations:

Operators should not provide identifiable data to third-parties without explicit consent from clients (we take the stance here that simply using the resolution service itself does not constitute consent).

Even when consent is granted operators should employ data minimization techniques such as those described in Section 5.2.1 if data is shared with third-parties.

Operators should consider including specific guidelines for the collection of aggregated and/or anonymized data for research purposes, within or outside of their own organization.

TODO: More on data for research vs operations... how to still motivate operators to share anonymized data?

TODO: Guidelines for when consent is granted?

TODO: Applies to server data handling too.. could operators offer alternatives services one that implies consent for data processing, one that doesn't?

6. DNS privacy policy and practice statement

6.1. Recommended contents of a DPPPS

1 Policy

1.1 Recommendations. This section should explain, with reference to section Section 5 of this document which recommendations the DNS privacy service employs.

1.2 Data handling. This section should explain, with reference to section Section 5.2 of this document the policy for gathering and disseminating information collected by the DNS privacy service.

1.2.1 Specify clearly what data (including whether it is aggregated, pseudonymized or anonymized) is:

1.2.1.1 Collected and retained by the operator (and for how long)

1.2.1.2 Shared with partners

1.2.1.3 Shared, sold or rented to third-parties

1.2.2 Specify any exceptions to the above, for example technically malicious or anomalous behaviour

1.2.3 Declare any partners, third-party affiliations or sources of funding

1.2.4 Whether user DNS data is correlated or combined with any other personal information held by the operator

2 Practice. This section should explain the current operational practices of the service.

2.1 Specify any temporary or permanent deviations from the policy for operational reasons

2.2 With reference to section Section 5.1 provide specific details of which capabilities are provided on which address and ports

2.3 With reference to section Section 5.3 provide specific details of which capabilities are employed for upstream traffic from the server

2.4 Specify the authentication name to be used (if any) and if TLSA records are published (including options used in the TLSA records)

2.5 Specify the SPKI pinsets to be used (if any) and policy for rolling keys

2.6 Provide a contact email address for the service

6.2. Current policy and privacy statements

NOTE: An analysis of these statements will clearly only provide a snapshot at the time of writing. It is included in this version of the draft to provide a basis for the assessment of the contents of the DPPPS and is expected to be removed or substantially re-worked in a future version.

6.2.1. Quad9

UDP/TCP and TLS (port 853) service provided on two addresses:

- o 'Secure': 9.9.9.9, 149.112.112.112, 2620:fe::fe, 2620:fe::9
- o 'Unsecured': 9.9.9.10, 149.112.112.10, 2620:fe::10

Policy:

- o <<https://www.quad9.net/policy/>>
- o <<https://www.quad9.net/privacy/>>
- o <<https://www.quad9.net/faq/>>

6.2.2. Cloudflare

UDP/TCP and TLS (port 853) service provided on 1.1.1.1, 1.0.0.1, 2606:4700:4700::1111 and 2606:4700:4700::1001.

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/privacy-policy/>>

DoH provided on: <<https://cloudflare-dns.com/dns-query>>

Policy:

- o <<https://developers.cloudflare.com/1.1.1.1/commitment-to-privacy/privacy-policy/firefox/>>

Tor endpoint: <<https://dns4torpnlfs2ifuz2s2yf3fc7rdmsbhm6rw75euj35pac6ap25zgqad.onion>>.

6.2.3. Google

UDP/TCP service provided on 8.8.8.8, 8.8.4.4, 2001:4860:4860::8888 and 2001:4860:4860::8844.

Policy: <<https://developers.google.com/speed/public-dns/privacy>>

6.2.4. OpenDNS

UDP/TCP service provided on 208.67.222.222 and 208.67.220.220 (no IPv6).

We could find no specific privacy policy for the DNS resolution, only a general one from Cisco that seems focussed on websites.

Policy: <<https://www.cisco.com/c/en/us/about/legal/privacy-full.html>>

6.2.5. Comparison

The following tables provides a high-level comparison of the policy and practice statements above and also some observations of practice measured at dnsprivacy.org [7]. The data is not exhaustive and has not been reviewed or confirmed by the operators.

A question mark indicates no clear statement or data could be located on the issue. A dash indicates the category is not applicable to the service.

Table showing comparison of operators policies [8]

Table showing comparison of operators practices [9]

NOTE: Review and correction of any inaccuracies in the table would be much appreciated.

6.3. Enforcement/accountability

Transparency reports may help with building user trust that operators adhere to their policies and practices.

Independent monitoring should be performed where possible of:

- o ECS, QNAME minimization, EDNS(0) padding, etc.
- o Filtering
- o Uptime

7. IANA considerations

None

8. Security considerations

TODO: e.g. New issues for DoS defence, server admin policies

9. Acknowledgements

Many thanks to Amelia Andersdotter for a very thorough review of the first draft of this document. Thanks also to John Todd for discussions on this topic, and to Stephane Bortzmeyer for review.

Sara Dickinson thanks the Open Technology Fund for a grant to support the work on this document.

10. Contributors

The below individuals contributed significantly to the document:

John Dickinson
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Jim Hague
Sinodun Internet Technologies
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

11. Changelog

draft-dickinson-dprive-bcp-op-01

- o Update reference to RFC7626 to draft-bortzmeyer-rfc7626-bis
- o Fix a few typos

draft-dickinson-dprive-bcp-op-00

Name change to add dprive. Differences to draft-dickinson-bcp-op-00:

- o Reworked the Terminology, Introduction and Scope

- o Added Document section
 - o Reworked the Recommendations section to describe threat mitigations, optimizations and other options. Split the recommendations up into 3 subsections: on the wire, at rest and upstream
 - o Added much more information on data handling and IP address pseudonymization and anonymization
 - o Added more details and comparison of some existing policy/privacy policies
 - o Applied virtually all of Amelia Andersdotter's suggested changes.
- draft-dickinson-bcp-op-00
- o Initial commit

12. References

12.1. Normative References

- [I-D.ietf-dnsop-terminology-bis]
Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", draft-ietf-dnsop-terminology-bis-11 (work in progress), July 2018.
- [I-D.ietf-doh-dns-over-https]
Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", draft-ietf-doh-dns-over-https-12 (work in progress), June 2018.
- [I-D.ietf-dprive-padding-policy]
Mayrhofer, A., "Padding Policy for EDNS(0)", draft-ietf-dprive-padding-policy-05 (work in progress), April 2018.
- [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>>.
- [RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <<https://www.rfc-editor.org/info/rfc5077>>.

- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <<https://www.rfc-editor.org/info/rfc6265>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC7830] Mayrhofer, A., "The EDNS(0) Padding Option", RFC 7830, DOI 10.17487/RFC7830, May 2016, <<https://www.rfc-editor.org/info/rfc7830>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC7873] Eastlake 3rd, D. and M. Andrews, "Domain Name System (DNS) Cookies", RFC 7873, DOI 10.17487/RFC7873, May 2016, <<https://www.rfc-editor.org/info/rfc7873>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.

12.2. Informative References

- [I-D.bortzmeyer-dprive-rfc7626-bis]
Bortzmeyer, S. and S. Dickinson, "DNS Privacy Considerations", draft-bortzmeyer-dprive-rfc7626-bis-00 (work in progress), July 2018.

- [I-D.ietf-dnsop-dns-capture-format]
Dickinson, J., Hague, J., Dickinson, S., Manderson, T.,
and J. Bond, "C-DNS: A DNS Packet Capture Format", draft-
ietf-dnsop-dns-capture-format-07 (work in progress), May
2018.
- [I-D.ietf-dnsop-dns-tcp-requirements]
Kristoff, J. and D. Wessels, "DNS Transport over TCP -
Operational Requirements", draft-ietf-dnsop-dns-tcp-
requirements-02 (work in progress), May 2018.
- [I-D.ietf-dnsop-session-signal]
Bellis, R., Cheshire, S., Dickinson, J., Dickinson, S.,
Lemon, T., and T. Pusateri, "DNS Stateful Operations",
draft-ietf-dnsop-session-signal-11 (work in progress),
July 2018.
- [I-D.ietf-tls-dnssec-chain-extension]
Shore, M., Barnes, R., Huque, S., and W. Toorop, "A DANE
Record and DNSSEC Authentication Chain Extension for TLS",
draft-ietf-tls-dnssec-chain-extension-07 (work in
progress), March 2018.
- [pcap] tcpdump.org, "PCAP", 2016, <<http://www.tcpdump.org/>>.
- [Pitfalls-of-DNS-Encryption]
Shulman, H., "Pretty Bad Privacy: Pitfalls of DNS
Encryption", 2014, <[https://www.ietf.org/mail-archive/web/
dns-privacy/current/pdfWqAIUmEl47.pdf](https://www.ietf.org/mail-archive/web/dns-privacy/current/pdfWqAIUmEl47.pdf)>.
- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization
Support", RFC 6235, DOI 10.17487/RFC6235, May 2011,
<<https://www.rfc-editor.org/info/rfc6235>>.
- [RFC6841] Ljunggren, F., Eklund Lowinder, AM., and T. Okubo, "A
Framework for DNSSEC Policies and DNSSEC Practice
Statements", RFC 6841, DOI 10.17487/RFC6841, January 2013,
<<https://www.rfc-editor.org/info/rfc6841>>.
- [RFC6873] Salgueiro, G., Gurbani, V., and A. Roach, "Format for the
Session Initiation Protocol (SIP) Common Log Format
(CLF)", RFC 6873, DOI 10.17487/RFC6873, February 2013,
<<https://www.rfc-editor.org/info/rfc6873>>.
- [RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use
Domain Name", RFC 7686, DOI 10.17487/RFC7686, October
2015, <<https://www.rfc-editor.org/info/rfc7686>>.

- [RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <<https://www.rfc-editor.org/info/rfc7706>>.
- [RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <<https://www.rfc-editor.org/info/rfc7766>>.
- [RFC7828] Wouters, P., Abley, J., Dickinson, S., and R. Bellis, "The edns-tcp-keepalive EDNS0 Option", RFC 7828, DOI 10.17487/RFC7828, April 2016, <<https://www.rfc-editor.org/info/rfc7828>>.
- [RFC7871] Contavalli, C., van der Gaast, W., Lawrence, D., and W. Kumari, "Client Subnet in DNS Queries", RFC 7871, DOI 10.17487/RFC7871, May 2016, <<https://www.rfc-editor.org/info/rfc7871>>.
- [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <<https://www.rfc-editor.org/info/rfc8094>>.
- [RFC8198] Fujiwara, K., Kato, A., and W. Kumari, "Aggressive Use of DNSSEC-Validated Cache", RFC 8198, DOI 10.17487/RFC8198, July 2017, <<https://www.rfc-editor.org/info/rfc8198>>.

12.3. URIs

- [1] <https://nginx.org/>
- [2] <https://www.haproxy.org/>
- [3] <https://kb.isc.org/article/AA-01386/0/DNS-over-TLS.html>
- [4] <https://doi.org/10.1145/3182660>
- [5] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/ip_techniques_table.svg
- [6] <https://pdfs.semanticscholar.org/7b34/12c951cebe71cd2cddac5fda164fb2138a44.pdf>
- [7] <https://dnsprivacy.org/jenkins/job/dnsprivacy-monitoring/>

- [8] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/policy_table.svg
- [9] https://github.com/Sinodun/draft-dprive-bcp-op/blob/master/draft-01/practice_table.svg
- [10] <https://support.google.com/analytics/answer/2763052?hl=en>
- [11] <https://www.conversionworks.co.uk/blog/2017/05/19/anonymize-ip-geo-impact-test/>
- [12] <https://github.com/edmonds/pdns/blob/master/pdns/dnswasher.cc>
- [13] <http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html>
- [14] <http://an.kaist.ac.kr/~sbmoon/paper/intl-journal/2004-cn-anon.pdf>
- [15] <https://www.cc.gatech.edu/computing/Telecomm/projects/cryptopan/>
- [16] http://mharvan.net/talks/noms-ip_anon.pdf
- [17] <https://medium.com/@bert.hubert/on-ip-address-encryption-security-analysis-with-respect-for-privacy-dabel201b476>
- [18] <https://github.com/PowerDNS/ipcipher>
- [19] <https://github.com/veorq/ipcrypt>
- [20] <https://www.ietf.org/mail-archive/web/cfrg/current/msg09494.html>
- [21] <https://tncl8.geant.org/core/presentation/127>

Appendix A. Documents

This section provides an overview of some DNS privacy related documents, however, this is neither an exhaustive list nor a definitive statement on the characteristic of the document.

A.1. Potential increases in DNS privacy

These documents are limited in scope to communications between stub clients and recursive resolvers:

- o 'Specification for DNS over Transport Layer Security (TLS)' [RFC7858], referred to here as 'DNS-over-TLS'.

- o 'DNS over Datagram Transport Layer Security (DTLS)' [RFC8094], referred to here as 'DNS-over-DTLS'. Note that this document has the Category of Experimental.
- o 'DNS Queries over HTTPS (DoH)' [I-D.ietf-doh-dns-over-https] referred to here as DoH.
- o 'Usage Profiles for DNS over TLS and DNS over DTLS' [RFC8310]
- o 'The EDNS(0) Padding Option' [RFC7830] and 'Padding Policy for EDNS(0)' [I-D.ietf-dprive-padding-policy]

These documents apply to recursive to authoritative DNS but are relevant when considering the operation of a recursive server:

- o 'DNS Query Name minimization to Improve Privacy' [RFC7816] referred to here as 'QNAME minimization'

A.2. Potential decreases in DNS privacy

These documents relate to functionality that could provide increased tracking of user activity as a side effect:

- o 'Client Subnet in DNS Queries' [RFC7871]
- o 'Domain Name System (DNS) Cookies' [RFC7873])
- o 'Transport Layer Security (TLS) Session Resumption without Server-Side State' [RFC5077] referred to here as simply TLS session resumption.
- o 'A DNS Packet Capture Format' [I-D.ietf-dnsop-dns-capture-format]
- o Passive DNS [I-D.ietf-dnsop-terminology-bis]

Note that depending on the specifics of the implementation [I-D.ietf-doh-dns-over-https] may also provide increased tracking.

A.3. Related operational documents

- o 'DNS Transport over TCP - Implementation Requirements' [RFC7766]
- o 'Operational requirements for DNS-over-TCP' [I-D.ietf-dnsop-dns-tcp-requirements]
- o 'The edns-tcp-keepalive EDNS0 Option' [RFC7828]
- o 'DNS Stateful Operations' [I-D.ietf-dnsop-session-signal]

Appendix B. IP address techniques

Data minimization methods may be categorized by the processing used and the properties of their outputs. The following builds on the categorization employed in [RFC6235]:

- o Format-preserving. Normally when encrypting, the original data length and patterns in the data should be hidden from an attacker. Some applications of de-identification, such as network capture de-identification, require that the de-identified data is of the same form as the original data, to allow the data to be parsed in the same way as the original.
- o Prefix preservation. Values such as IP addresses and MAC addresses contain prefix information that can be valuable in analysis, e.g. manufacturer ID in MAC addresses, subnet in IP addresses. Prefix preservation ensures that prefixes are de-identified consistently; e.g. if two IP addresses are from the same subnet, a prefix preserving de-identification will ensure that their de-identified counterparts will also share a subnet. Prefix preservation may be fixed (i.e. based on a user selected prefix length identified in advance to be preserved) or general.
- o Replacement. A one-to-one replacement of a field to a new value of the same type, for example using a regular expression.
- o Filtering. Removing (and thus truncating) or replacing data in a field. Field data can be overwritten, often with zeros, either partially (grey marking) or completely (black marking).
- o Generalization. Data is replaced by more general data with reduced specificity. One example would be to replace all TCP/UDP port numbers with one of two fixed values indicating whether the original port was ephemeral (≥ 1024) or non-ephemeral (> 1024). Another example, precision degradation, reduces the accuracy of e.g. a numeric value or a timestamp.
- o Enumeration. With data from a well-ordered set, replace the first data item data using a random initial value and then allocate ordered values for subsequent data items. When used with timestamp data, this preserves ordering but loses precision and distance.
- o Reordering/shuffling. Preserving the original data, but rearranging its order, often in a random manner.
- o Random substitution. As replacement, but using randomly generated replacement values.

- o Cryptographic permutation. Using a permutation function, such as a hash function or cryptographic block cipher, to generate a replacement de-identified value.

B.1. Google Analytics non-prefix filtering

Since May 2010, Google Analytics has provided a facility [10] that allows website owners to request that all their users IP addresses are anonymized within Google Analytics processing. This very basic anonymization simply sets to zero the least significant 8 bits of IPv4 addresses, and the least significant 80 bits of IPv6 addresses. The level of anonymization this produces is perhaps questionable. There are some analysis results [11] which suggest that the impact of this on reducing the accuracy of determining the user's location from their IP address is less than might be hoped; the average discrepancy in identification of the user city for UK users is no more than 17%.

Anonymization: Format-preserving, Filtering (grey marking).

B.2. dnswasher

Since 2006, PowerDNS have included a de-identification tool dnswasher [12] with their PowerDNS product. This is a PCAP filter that performs a one-to-one mapping of end user IP addresses with an anonymized address. A table of user IP addresses and their de-identified counterparts is kept; the first IPv4 user addresses is translated to 0.0.0.1, the second to 0.0.0.2 and so on. The de-identified address therefore depends on the order that addresses arrive in the input, and running over a large amount of data the address translation tables can grow to a significant size.

Anonymization: Format-preserving, Enumeration.

B.3. Prefix-preserving map

Used in TCPdpriv [13], this algorithm stores a set of original and anonymised IP address pairs. When a new IP address arrives, it is compared with previous addresses to determine the longest prefix match. The new address is anonymized by using the same prefix, with the remainder of the address anonymized with a random value. The use of a random value means that TCPdpriv is not deterministic; different anonymized values will be generated on each run. The need to store previous addresses means that TCPdpriv has significant and unbounded memory requirements, and because of the need to allocated anonymized addresses sequentially cannot be used in parallel processing.

Anonymization: Format-preserving, prefix preservation (general).

B.4. Cryptographic Prefix-Preserving Pseudonymisation

Cryptographic prefix-preserving pseudonymisation was originally proposed as an improvement to the prefix-preserving map implemented in TCPdpriv, described in Xu et al. [14] and implemented in the Crypto-PAN tool [15]. Crypto-PAN is now frequently used as an acronym for the algorithm. Initially it was described for IPv4 addresses only; extension for IPv6 addresses was proposed in Harvan & Schoenwaelder [16] and implemented in snmpdump. This uses a cryptographic algorithm rather than a random value, and thus pseudonymity is determined uniquely by the encryption key, and is deterministic. It requires a separate AES encryption for each output bit, so has a non-trivial calculation overhead. This can be mitigated to some extent (for IPv4, at least) by pre-calculating results for some number of prefix bits.

Pseudonymization: Format-preserving, prefix preservation (general).

B.5. Top-hash Subtree-replicated Anonymisation

Proposed in Ramaswamy & Wolf, Top-hash Subtree-replicated Anonymisation (TSA) originated in response to the requirement for faster processing than Crypto-PAN. It used hashing for the most significant byte of an IPv4 address, and a pre-calculated binary tree structure for the remainder of the address. To save memory space, replication is used within the tree structure, reducing the size of the pre-calculated structures to a few Mb for IPv4 addresses. Address pseudonymization is done via hash and table lookup, and so requires minimal computation. However, due to the much increased address space for IPv6, TSA is not memory efficient for IPv6.

Pseudonymization: Format-preserving, prefix preservation (general).

B.6. ipcipher

A recently-released proposal from PowerDNS [17], ipcipher [18] is a simple pseudonymization technique for IPv4 and IPv6 addresses. IPv6 addresses are encrypted directly with AES-128 using a key (which may be derived from a passphrase). IPv4 addresses are similarly encrypted, but using a recently proposed encryption ipcrypt [19] suitable for 32bit block lengths. However, the author of ipcrypt has since indicated [20] that it has low security, and further analysis has revealed it is vulnerable to attack.

Pseudonymization: Format-preserving, cryptographic permutation.

B.7. Bloom filters

van Rijswijk-Deij et al. [21] have recently described work using Bloom filters to categorize query traffic and record the traffic as the state of multiple filters. The goal of this work is to allow operators to identify so-called Indicators of Compromise (IOCs) originating from specific subnets without storing information about, or be able to monitor the DNS queries of an individual user. By using a Bloom filter, it is possible to determine with a high probability if, for example, a particular query was made, but the set of queries made cannot be recovered from the filter. Similarly, by mixing queries from a sufficient number of users in a single filter, it becomes practically impossible to determine if a particular user performed a particular query. Large numbers of queries can be tracked in a memory-efficient way. As filter status is stored, this approach cannot be used to regenerate traffic, and so cannot be used with tools used to process live traffic.

Anonymized: Generalization.

Authors' Addresses

Sara Dickinson
Sinodun IT
Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Email: sara@sinodun.com

Benno J. Overeinder
NLnet Labs
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: benno@nlnetLabs.nl

Roland M. van Rijswijk-Deij
SURFnet bv
PO Box 19035
Utrecht 3501 DA Utrecht
The Netherlands

Email: roland.vanrijswijk@surfnet.nl

Allison Mankin
Salesforce

Email: allison.mankin@gmail.com

TSVWG
Internet-Draft
Intended status: Informational
Expires: December 14, 2018

G. Fairhurst
University of Aberdeen
C.S. Perkins
University of Glasgow
June 14, 2018

The Impact of Transport Header Confidentiality on Network Operation and
Evolution of the Internet
draft-fairhurst-tsvwg-transport-encrypt-09

Abstract

This document describes implications of applying end-to-end encryption at the transport layer. It identifies in-network uses of transport layer header information. It then reviews the implications of developing end-to-end transport protocols that use authentication to protect the integrity of transport information or encryption to provide confidentiality of the transport protocol header and expected implications of transport protocol design and network operation. Since transport measurement and analysis of the impact of network characteristics have been important to the design of current transport protocols, it also considers the impact on transport and application evolution.

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 14, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Current uses of Transport Headers within the Network	8
2.1. Observing Transport Information in the Network	9
2.1.1. Flow Identification	9
2.1.2. Metrics derived from Transport Layer Headers	10
2.1.3. Metrics derived from Network Layer Headers	13
2.2. Transport Measurement	14
2.2.1. Point of Measurement	15
2.2.2. Use by Operators to Plan and Provision Networks	15
2.2.3. Service Performance Measurement	16
2.2.4. Measuring Transport to Support Network Operations	16
2.3. Use for Network Diagnostics and Troubleshooting	17
2.3.1. Examples of measurements	18
2.4. Observing Headers to Implement Network Policy	19
3. Encryption and Authentication of Transport Headers	19
3.1. Authenticating the Transport Protocol Header	21
3.2. Encrypting the Transport Payload	21
3.3. Encrypting the Transport Header	21
3.4. Authenticating Transport Information and Selectively Encrypting the Transport Header	22
3.5. Optional Encryption of Header Information	22
4. Addition of Transport Information to Network-Layer Protocol Headers	22
5. Implications of Protecting the Transport Headers	23
5.1. Independent Measurement	23
5.2. Characterising "Unknown" Network Traffic	24
5.3. Accountability and Internet Transport Protocols	25
5.4. Impact on Research, Development and Deployment	25
6. Conclusions	27
7. Acknowledgements	29
8. Security Considerations	29
9. IANA Considerations	31
10. References	31
10.1. Normative References	31
10.2. Informative References	31
Appendix A. Revision information	36
Authors' Addresses	37

1. Introduction

This document describes implications of applying end-to-end encryption at the transport layer. It reviews the implications of developing end-to-end transport protocols that use encryption to provide confidentiality of the transport protocol header and expected implications of transport protocol design and network operation. It also considers anticipated implications on transport and application evolution.

The transport layer provides end-to-end interactions between endpoints (processes) using an Internet path. Transport protocols layer directly over the network-layer service and are sent in the payload of network-layer packets. They support end-to-end communication between applications, supported by higher-layer protocols, running on the end systems (or transport endpoints). This simple architectural view hides one of the core functions of the transport, however, to discover and adapt to the properties of the Internet path that is currently being used. The design of Internet transport protocols is as much about trying to avoid the unwanted side effects of congestion on a flow and other capacity-sharing flows, avoiding congestion collapse, adapting to changes in the path characteristics, etc., as it is about end-to-end feature negotiation, flow control and optimising for performance of a specific application.

To achieve stable Internet operations the IETF transport community has to date relied heavily on measurement and insights of the network operations community to understand the trade-offs, and to inform selection of appropriate mechanisms, to ensure a safe, reliable, and robust Internet (e.g., [RFC1273]). In turn, the network operations community relies on being able to understand the pattern and requirements of traffic passing over the Internet, both in aggregate and at the flow level.

There are many motivations for deploying encrypted transports [RFC7624] (i.e., transport protocols that use encryption to provide confidentiality of some or all of the transport-layer header information), and encryption of transport payloads (i.e. confidentiality of the payload data). The increasing public concerns about the interference with Internet traffic have led to a rapidly expanding deployment of encryption to protect end-user privacy, in protocols like QUIC [I-D.ietf-quic-transport], but also expected to form a basis of future protocol designs.

Some network operators and access providers, have come to rely on the in-network measurement of transport properties and the functionality provided by middleboxes to both support network operations and enhance performance. There can therefore be implications when working with encrypted transport protocols that hide transport header information from the network. These present architectural challenges and considerations in the way transport protocols are designed, and ability to characterise and compare different transport solutions [Measure], Section 2.2. Implementations of network devices are encouraged to avoid side-effects when protocols are updated. Introducing cryptographic integrity checks to header fields can also prevent undetected manipulation of the field by network devices, or undetected addition of information to a packet. However, this does not prevent inspection of the information by a device on path, and it is possible that such devices could develop mechanisms that rely on the presence of such a field, or a known value in the field.

Reliance on the presence and semantics of specific header information leads to ossification: An endpoint could be required to supply a specific header to receive the network service that it desires. In some cases, this could be benign or advantageous to the protocol (e.g., recognising the start of a connection, or explicitly exposing protocol information can be expected to provide more consistent decisions by on-path devices than the use of diverse methods to infer semantics from other flow properties). In some cases, this is not beneficial (e.g., a mechanism implemented in a network device, such as a firewall, that required a header field to have only a specific known set of values could prevent the device from forwarding packets using a different version of a protocol that introduces a new feature that changes the value present in this field, preventing evolution of the protocol).

A protocol design that uses header encryption can provide confidentiality of some or all of the protocol header information. This prevents an on-path device from knowledge of the header field. It therefore prevents mechanisms being built that directly rely on the information or seeks to imply semantics of an exposed header field. Using encryption to provide confidentiality of the transport layer brings some well-known privacy and security benefits and can therefore help reduce ossification of the transport layer. In particular, it is important that protocols either do not expose information where the usage may change in future protocols, or that methods that utilise the information are robust to potential changes as protocols evolve over time. To avoid unwanted inspection, a protocol could also intentionally vary the format and value of header fields (sometimes known as Greasing [I-D.thomson-quick-grease]).

However, while encryption hides the protocol header information, it does not prevent ossification of the network service: People seeking understanding of network traffic could come to rely on pattern inferences and other heuristics as the basis for network decision and to derive measurement data, creating new dependencies on the transport protocol.

A level of ossification of the transport header can offer trade-offs around authentication, and confidentiality of transport protocol headers and has the potential to explicitly support for other uses of this header information. For example, a design that provides confidentiality of protocol header information can impact the following activities that rely on measurement and analysis of traffic flows:

Network Operations and Research: Observable transport headers enable both operators and the research community to measure and analyse protocol performance, network anomalies, and failure pathologies.

This information can help inform capacity planning, and assist in determining the need for equipment and/or configuration changes by network operators.

The data can also inform Internet engineering research, and help in the development of new protocols, methodologies, and procedures. Concealing the transport protocol header information makes the stream performance unavailable to passive observers along the path, and likely leads to the development of alternative methods to collect or infer that data.

Providing confidentiality of the transport payload, but leaving some, or all, of the transport headers unencrypted, possibly with authentication, can provide the majority of the privacy and security benefits while allowing some measurement.

Protection from Denial of Service: Observable transport headers currently provide useful input to classify traffic and detect anomalous events (e.g., changes in application behaviour, distributed denial of service attacks). To be effective, this protection needs to be able to uniquely disambiguate unwanted traffic. An inability to separate this traffic using packet header information may result in less-efficient identification of unwanted traffic or development of different methods (e.g. rate-limiting of uncharacterised traffic).

Network Troubleshooting and Diagnostics: Encrypting transport header information eliminates the incentive for operators to troubleshoot what they cannot interpret. A flow experiencing packet loss or jitter looks like an unaffected flow when only observing network layer headers (if transport sequence numbers and flow identifiers are obscured). This limits understanding of the impact of packet loss or latency on the flows, or even localizing the network segment causing the packet loss or latency. Encrypted traffic may imply "don't touch" to some, and could limit a trouble-shooting response to "can't help, no trouble found". The additional mechanisms that will need to be introduced to help reconstruct transport-level metrics add complexity and operational costs (e.g., in deploying additional functions in equipment or adding traffic overhead).

Network Traffic Analysis: Hiding transport protocol header information can make it harder to determine which transport protocols and features are being used across a network segment and to measure trends in the pattern of usage. This could impact the ability for an operator to anticipate the need for network upgrades and roll-out. It can also impact the on-going traffic engineering activities performed by operators (such as determining which parts of the path contribute delay, jitter or loss). While the impact may, in many cases, be small there are scenarios where operators directly support particular services (e.g., to troubleshoot issues relating to Quality of Service, QoS; the ability to perform fast re-routing of critical traffic, or support to mitigate the characteristics of specific radio links). The more complex the underlying infrastructure the more important this impact.

Open and Verifiable Network Data: Hiding transport protocol header information can reduce the range of actors that can capture useful measurement data. For example, one approach could be to employ an existing transport protocol that reveals little information (e.g., UDP), and perform traditional transport functions at higher layers protecting the confidentiality of transport information. Such a design, limits the information sources available to the Internet community to understand the operation of new transport protocols, so preventing access to the information necessary to inform design decisions and standardisation of the new protocols and related operational practices.

The cooperating dependence of network, application, and host to provide communication performance on the Internet is uncertain when only endpoints (i.e., at user devices and within service platforms) can observe performance, and performance cannot be

independently verified by all parties. The ability of other stakeholders to review code can help develop deeper insight. In the heterogeneous Internet, this helps extend the range of topologies, vendor equipment, and traffic patterns that are evaluated.

Independently captured data is important to help ensure the health of the research and development communities. It can provide input and test scenarios to support development of new transport protocol mechanisms, especially when this analysis can be based on the behaviour experienced in a diversity of deployed networks.

Independently verifiable performance metrics might also be important to demonstrate regulatory compliance in some jurisdictions, and provides an important basis for informing design decisions.

The last point leads us to consider the impact of hiding transport headers in the specification and development of protocols and standards. This has potential impact on:

- o Understanding Feature Interactions: An appropriate vantage point, coupled with timing information about traffic flows, provides a valuable tool for benchmarking equipment, functions, and/or configurations, and to understand complex feature interactions. An inability to observe transport protocol information can limit the ability to diagnose and explore interactions between features at different protocol layers, a side-effect of not allowing a choice of vantage point from which this information is observed.
- o Supporting Common Specifications: Transmission Control Protocol (TCP) is currently the predominant transport protocol used over Internet paths. Its many variants have broadly consistent approaches to avoiding congestion collapse, and to ensuring the stability of the Internet. Increased use of transport layer encryption can overcome ossification, allowing deployment of new transports and different types of congestion control. This flexibility can be beneficial, but it can come at the cost of fragmenting the ecosystem. There is little doubt that developers will try to produce high quality transports for their intended target uses, but it is not clear there are sufficient incentives to ensure good practice that benefits the wide diversity of requirements for the Internet community as a whole. Increased diversity, and the ability to innovate without public scrutiny, risks point solutions that optimise for specific needs, but accidentally disrupt operations of/in different parts of the network. The social contract that maintains the stability of the Internet relies on accepting common specifications, and on the ability to verify that others also conform.

- o Operational practice: Published transport specifications allow operators to check compliance. This can bring assurance to those operating networks, often avoiding the need to deploy complex techniques that routinely monitor and manage TCP/IP traffic flows (e.g. Avoiding the capital and operational costs of deploying flow rate-limiting and network circuit-breaker methods [RFC8084]). When it is not possible to observe transport header information, methods are still needed to confirm that the traffic produced conforms to the expectations of the operator or developer.
- o Restricting research and development: Hiding transport information can impede independent research into new mechanisms, measurement of behaviour, and development initiatives. Experience shows that transport protocols are complicated to design and complex to deploy, and that individual mechanisms need to be evaluated while considering other mechanisms, across a broad range of network topologies and with attention to the impact on traffic sharing the capacity. If this results in reduced availability of open data, it could eliminate the independent self-checks to the standardisation process that have previously been in place from research and academic contributors (e.g., the role of the IRTF ICCRG, and research publications in reviewing new transport mechanisms and assessing the impact of their experimental deployment)

In summary, there are trade offs. On the one hand, protocol designers have often ignored the implications of whether the information in transport header fields can or will be used by in-network devices, and the implications this places on protocol evolution. This motivates a design that provides confidentiality of the header information. On the other hand, it can be expected that a lack of visibility of transport header information can impact the ways that protocols are deployed, standardised, and their operational support. The choice of whether future transport protocols encrypt their protocol headers therefore needs to be taken based not solely on security and privacy considerations, but also taking into account the impact on operations, standards, and research. Any new Internet transport need to provide appropriate transport mechanisms and operational support to assure the resulting traffic can not result in persistent congestion collapse [RFC2914]. This document suggests that the balance between information exposed and concealed should be carefully considered when specifying new protocols.

2. Current uses of Transport Headers within the Network

Despite transport headers having end-to-end meaning, some of these transport headers have come to be used in various ways within the Internet. In response to pervasive monitoring [RFC7624] revelations and the IETF consensus that "Pervasive Monitoring is an Attack" [RFC7258], efforts are underway to increase encryption of Internet traffic,. Applying confidentiality to transport header fields would affect how protocol information is used [I-D.mm-wg-effect-encrypt]. To understand these implications, it is first necessary to understand how transport layer headers are currently observed and/or modified by middleboxes within the network.

Transport protocols can be designed to encrypt or authenticate transport header fields. Authentication at the transport layer can be used to detect any changes to an immutable header field that were made by a network device along a path. The intentional modification of transport headers by middleboxes (such as Network Address Translation, NAT, or Firewalls) is not considered. Common issues concerning IP address sharing are described in [RFC6269].

2.1. Observing Transport Information in the Network

If in-network observation of transport protocol headers is needed, this requires knowledge of the format of the transport header:

- o Flows need to be identified at the level required to perform the observation;
- o The protocol and version of the header need to be visible. As protocols evolve over time and there may be a need to introduce new transport headers. This may require interpretation of protocol version information or connection setup information;
- o The location and syntax of any observed transport headers needs to be known. IETF transport protocols can specify this information.

The following subsections describe various ways that observable transport information has been utilised.

2.1.1. Flow Identification

Transport protocol header information (together with information in the network header), has been used to identify a flow and the connection state of the flow, together with the protocol options being used. In some usages, a low-numbered (well-known) transport port number has been used to identify a protocol (although port

information alone is not sufficient to guarantee identification of a protocol, since applications can use arbitrary ports, multiple sessions can be multiplexed on a single port, and ports can be re-used by subsequent sessions).

Transport protocols, such as TCP and Stream Control Transport Protocol (SCTP) specify a standard base header that includes sequence number information and other data, with the possibility to negotiate additional headers at connection setup, identified by an option number in the transport header. UDP-based protocols can use, but sometimes do not use, well-known port numbers. Some flows can instead be identified by signalling protocols or through the use of magic numbers placed in the first byte(s) of the datagram payload.

Flow identification is a common function. For example, performed by measurement activities, QoS classification, firewalls, Denial of Service, DOS, prevention. It becomes more complex and less easily achieved when multiplexing is used at or above the transport layer.

2.1.2. Metrics derived from Transport Layer Headers

Some actors manage their portion of the Internet by characterizing the performance of link/network segments. Passive monitoring uses observed traffic to make inferences from transport headers to derive these measurements. A variety of open source and commercial tools have been deployed that utilise this information. The following metrics can be derived from transport header information:

Traffic Rate and Volume: Header information e.g., (sequence number, length) allows derivation of volume measures per-application, to characterise the traffic that uses a network segment or the pattern of network usage. This may be measured per endpoint or for an aggregate of endpoints (e.g., by an operator to assess subscriber usage). It can also be used to trigger measurement-based traffic shaping and to implement QoS support within the network and lower layers. Volume measures can be valuable for capacity planning (providing detail of trends rather than the volume per subscriber).

Loss Rate and Loss Pattern: Flow loss rate may be derived (e.g., from sequence number) and has been used as a metric for performance assessment and to characterise transport behaviour. Understanding the root cause of loss can help an operator determine whether this requires corrective action. Network operators have used the variation in patterns of loss as a key performance metric, utilising this to detect changes in the offered service.

There are various causes of loss, including: corruption of link frames (e.g., interference on a radio link), buffer overflow (e.g., due to congestion), policing (traffic management), buffer management (e.g., Active Queue Management, AQM [RFC7567]), inadequate provision of traffic preemption. Understanding flow loss rate requires either maintaining per flow packet counters or by observing sequence numbers in transport headers. Loss can be monitored at the interface level by devices in the network. It is often important to understand the conditions under which packet loss occurs. This usually requires relating loss to the traffic flowing on the network node/segment at the time of loss.

Observation of transport feedback information (observing loss reports, e.g., RTP Control Protocol (RTCP) [RFC3550], TCP SACK) can increase understanding of the impact of loss and help identify cases where loss may have been wrongly identified, or the transport did not require the lost packet. It is sometimes more important to understand the pattern of loss, than the loss rate, because losses can often occur as bursts, rather than randomly-timed events.

Throughput and Goodput: The throughput achieved by a flow can be determined even when a flow is encrypted, providing the individual flow can be identified. Goodput [RFC7928] is a measure of useful data exchanged (the ratio of useful/total volume of traffic sent by a flow). This requires ability to differentiate loss and retransmission of packets (e.g., by observing packet sequence numbers in the TCP or the Real Time Protocol, RTP, headers [RFC3550]).

Latency: Latency is a key performance metric that impacts application response time and user-perceived response time. It often indirectly impacts throughput and flow completion time. Latency determines the reaction time of the transport protocol itself, impacting flow setup, congestion control, loss recovery, and other transport mechanisms. The observed latency can have many components [Latency]. Of these, unnecessary/unwanted queuing in network buffers has often been observed as a significant factor. Once the cause of unwanted latency has been identified, this can often be eliminated.

To measure latency across a part of a path, an observation point can measure the experienced round trip time (RTT) using packet sequence numbers, and acknowledgements, or by observing header timestamp information. Such information allows an observation point in the network to determine not only the path RTT, but also to measure the upstream and downstream contribution to the RTT.

This has been used to locate a source of latency, e.g., by observing cases where the ratio of median to minimum RTT is large for a part of a path.

The service offered by operators can benefit from latency information to understand the impact of deployment and tune deployed services. Latency metrics are key to evaluating and deploying AQM [RFC7567], DiffServ [RFC2474], and Explicit Congestion Notification (ECN) [RFC3168] [RFC8087]. Measurements could identify excessively large buffers, indicating where to deploy or configure AQM. An AQM method is often deployed in combination with other techniques, such as scheduling [RFC7567] [I-D.ietf-aqm-fq-codel] and although parameter-less methods are desired [RFC7567], current methods [I-D.ietf-aqm-fq-codel] [I-D.ietf-aqm-codel] [I-D.ietf-aqm-pie] often cannot scale across all possible deployment scenarios.

Variation in delay: Some network applications are sensitive to small changes in packet timing. To assess the performance of such applications, it can be necessary to measure the variation in delay observed along a portion of the path [RFC3393] [RFC5481]. The requirements resemble those for the measurement of latency.

Flow Reordering: Significant flow reordering can impact time-critical applications and can be interpreted as loss by reliable transports. Many transport protocol techniques are impacted by reordering (e.g., triggering TCP retransmission, or re-buffering of real-time applications). Packet reordering can occur for many reasons (from equipment design to misconfiguration of forwarding rules). Since this impacts transport performance, network tools are needed to detect and measure unwanted/excessive reordering.

There have been initiatives in the IETF transport area to reduce the impact of reordering within a transport flow, possibly leading to a reduction in the requirements for preserving ordering. These have promise to simplify network equipment design as well as the potential to improve robustness of the transport service. Measurements of reordering can help understand the present level of reordering within deployed infrastructure, and inform decisions about how to progress such mechanisms.

Operational tools to detect mis-ordered packet flows and quantify the degree of reordering. Key performance indicators are retransmission rate, packet drop rate, sector utilisation level, a measure of reordering, peak rate, the ECN congestion experienced (CE) marking rate, etc.

Metrics have been defined that evaluate whether a network has maintained packet order on a packet-by-packet basis [RFC4737] and [RFC5236].

Techniques for measuring reordering typically observe packet sequence numbers. Some protocols provide in-built monitoring and reporting functions. Transport fields in the RTP header [RFC3550] [RFC4585] can be observed to derive traffic volume measurements and provide information on the progress and quality of a session using RTP. As with other measurement, metadata is often important to understand the context under which the data was collected, including the time, observation point, and way in which metrics were accumulated. The RTCP protocol directly reports some of this information in a form that can be directly visible in the network. A user of summary measurement data needs to trust the source of this data and the method used to generate the summary information.

2.1.3. Metrics derived from Network Layer Headers

Some transport information is made visible in the network-layer protocol header. These header fields are not encrypted and have been utilised to make flow observations.

Use of IPv6 Network-Layer Flow Label: Endpoints are encouraged expose flow information in the IPv6 Flow Label field of the network-layer header (e.g., [RFC8085]). This can be used to inform network-layer queuing, forwarding (e.g., for Equal Cost Multi-Path, ECMP, routing, and Link Aggregation, LAG). This can provide useful information to assign packets to flows in the data collected by measurement campaigns. Although important to characterising a path, it does not directly provide performance data.

Use Network-Layer Differentiated Services Code Point: Applications can expose their delivery expectations to the network by setting the Differentiated Services Code Point (DSCP) field of IPv4 and IPv6 packets. This can be used to inform network-layer queuing and forwarding, and can also provide information on the relative importance of packet information collected by measurement campaigns, but does not directly provide performance data.

This field provides explicit information that can be used in place of inferring traffic requirements (e.g., by inferring QoS requirements from port information via a multi-field classifier). The DSCP value can therefore impact the quality of experience for a flow. Observations of service performance need to consider this field when a network path has support for differentiated service treatment.

Use of Explicit Congestion Marking: ECN [RFC3168] is an optional transport mechanism that uses a code point in the network-layer header. Use of ECN can offer gains in terms of increased

throughput, reduced delay, and other benefits when used over a path that includes equipment that supports an AQM method that performs Congestion Experienced (CE) marking of IP packets [RFC8087].

ECN exposes the presence of congestion on a network path to the transport and network layer. The reception of CE-marked packets can therefore be used to monitor the presence and estimate the level of incipient congestion on the upstream portion of the path from the point of observation (Section 2.5 of [RFC8087]). Because ECN marks are carried in the IP protocol header, it is much easier to measure ECN than to measure packet loss. However, interpreting the marking behaviour (i.e., assessing congestion and diagnosing faults) requires context from the transport layer (path RTT, visibility of loss - that could be due to queue overflow, congestion response, etc) [RFC7567].

Some ECN-capable network devices can provide richer (more frequent and fine-grained) indication of their congestion state. Setting congestion marks proportional to the level of congestion (e.g., Data Center TCP, DCTP [RFC8257], and Low Latency Low Loss Scalable throughput, L4S, [I-D.ietf-tsvwg-l4s-arch]).

Use of ECN requires a transport to feed back reception information on the path towards the data sender. Exposure of this Transport ECN feedback provides an additional powerful tool to understand ECN-enabled AQM-based networks [RFC8087].

AQM and ECN offer a range of algorithms and configuration options, it is therefore important for tools to be available to network operators and researchers to understand the implication of configuration choices and transport behaviour as use of ECN increases and new methods emerge [RFC7567] [RFC8087]. ECN-monitoring is expected to become important as AQM is deployed that supports ECN [RFC8087].

2.2. Transport Measurement

The common language between network operators and application/content providers/users is packet transfer performance at a layer that all can view and analyse. For most packets, this has been transport layer, until the emergence of QUIC, with the obvious exception of Virtual Private Networks (VPNs) and IPsec.

When encryption conceals more layers in each packet, people seeking understanding of the network operation rely more on pattern inferences and other heuristics reliance on pattern inferences and accuracy suffers. For example, the traffic patterns between server

and browser are dependent on browser supplier and version, even when the sessions use the same server application (e.g., web e-mail access). It remains to be seen whether more complex inferences can be mastered to produce the same monitoring accuracy (see section 2.1.1 of [I-D.mm-wg-effect-encrypt]).

When measurement datasets are made available by servers or client endpoints, additional metadata, such as the state of the network, is often required to interpret this data. Collecting and coordinating such metadata is more difficult when the observation point is at a different location to the bottleneck/device under evaluation.

Packet sampling techniques can be used to scale the processing involved in observing packets on high rate links. This exports only the packet header information of (randomly) selected packets. The utility of these measurements depends on the type of bearer and number of mechanisms used by network devices. Simple routers are relatively easy to manage, a device with more complexity demands understanding of the choice of many system parameters. This level of complexity exists when several network methods are combined.

This section discusses topics concerning observation of transport flows, with a focus on transport measurement.

2.2.1. Point of Measurement

Often measurements can only be understood in the context of the other flows that share a bottleneck. A simple example is monitoring of AQM. For example, FQ-CODEL [I-D.ietf-aqm-fq-codel], combines sub queues (statistically assigned per flow), management of the queue length (CODEL), flow-scheduling, and a starvation prevention mechanism. Usually such algorithms are designed to be self-tuning, but current methods typically employ heuristics that can result in more loss under certain path conditions (e.g., large RTT, effects of multiple bottlenecks [RFC7567]).

In-network measurements can distinguish between upstream and downstream metrics with respect to a measurement point. These are particularly useful for locating the source of problems or to assess the performance of a network segment or a particular device configuration. By correlating observations of headers at multiple points along the path (e.g., at the ingress and egress of a network segment), an observer can determine the contribution of a portion of the path to an observed metric (to locate a source of delay, jitter, loss, reordering, congestion marking, etc.).

2.2.2. Use by Operators to Plan and Provision Networks

Traffic measurements (e.g., traffic volume, loss, latency) is used by operators to help plan deployment of new equipment and configurations in their networks. Data is also important to equipment vendors who need to understand traffic trends and patterns of usage as inputs to decisions about planning products and provisioning for new deployments. This measurement information can also be correlated with billing information when this is also collected by an operator.

A network operator supporting traffic that uses transport header encryption may not have access to per-flow measurement data. Trends in aggregate traffic can be observed and can be related to the endpoint addresses being used, but it may not be possible to correlate patterns in measurements with changes in transport protocols (e.g., the impact of changes in introducing a new transport protocol mechanism). This increases the dependency on other indirect sources of information to inform planning and provisioning.

2.2.3. Service Performance Measurement

Traffic measurements (e.g., traffic volume, loss, latency) can be used by various actors to help analyse the performance offered to the users of a network segment, and inform operational practice.

While active measurements may be used in-network, passive measurements can have advantages in terms of eliminating unproductive test traffic, reducing the influence of test traffic on the overall traffic mix, and the ability to choose the point of measurement Section 2.2.1. However, passive measurements may rely on observing transport headers.

2.2.4. Measuring Transport to Support Network Operations

Information provided by tools observing transport headers can help determine whether mechanisms are needed in the network to prevent flows from acquiring excessive network capacity. Operators can implement operational practices to manage traffic flows (e.g., to prevent flows from acquiring excessive network capacity under severe congestion) by deploying rate-limiters, traffic shaping or network transport circuit breakers [RFC8084].

Congestion Control Compliance of Traffic: Congestion control is a key transport function [RFC2914]. Many network operators implicitly accept that TCP traffic to comply with a behaviour that is acceptable for use in the shared Internet. TCP algorithms have been continuously improved over decades, and they have reached a level of efficiency and correctness that custom application-layer mechanisms will struggle to easily duplicate [RFC8085].

A standards-compliant TCP stack provides congestion control may therefore be judged safe for use across the Internet. Applications developed on top of well-designed transports can be expected to appropriately control their network usage, reacting when the network experiences congestion, by back-off and reduce the load placed on the network. This is the normal expected behaviour for IETF-specified transport (e.g., TCP and SCTP).

However, when anomalies are detected, tools can interpret the transport protocol header information to help understand the impact of specific transport protocols (or protocol mechanisms) on the other traffic that shares a network. An observation in the network can gain understanding of the dynamics of a flow and its congestion control behaviour. Analysing observed packet sequence numbers can be used to help build confidence that an application flow backs-off its share of the network load in the face of persistent congestion, and hence to understand whether the behaviour is appropriate for sharing limited network capacity. For example, it is common to visualise plots of TCP sequence numbers versus time for a flow to understand how a flow shares available capacity, deduce its dynamics in response to congestion, etc.

Congestion Control Compliance for UDP traffic UDP provides a minimal message-passing datagram transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and other protocols that choose to use UDP as a transport are required to employ mechanisms to prevent congestion collapse, avoid unacceptable contributions to jitter/latency, and to establish an acceptable share of capacity with concurrent traffic [RFC8085].

A network operator needs tools to understand if datagram flows comply with congestion control expectations and therefore whether there is a need to deploy methods such as rate-limiters, transport circuit breakers or other methods to enforce acceptable usage for the offered service.

UDP flows that expose a well-known header by specifying the format of header fields can allow information to be observed to gain understanding of the dynamics of a flow and its congestion control behaviour. For example, tools exist to monitor various aspects of the RTP and RTCP header information of real-time flows (see Section 2.1.2).

2.3. Use for Network Diagnostics and Troubleshooting

Transport header information can be useful for a variety of operational tasks [I-D.mm-wg-effect-encrypt]: to diagnose network problems, assess network provider performance, evaluate equipment/protocol performance, capacity planning, management of security threats (including denial of service), and responding to user performance questions. Sections 3.1.2 and 5 of [I-D.mm-wg-effect-encrypt] provide further examples. These tasks seldom involve the need to determine the contents of the transport payload, or other application details.

A network operator supporting traffic that uses transport header encryption can see only encrypted transport headers. This prevents deployment of performance measurement tools that rely on transport protocol information. Choosing to encrypt all the information reduces the operator's ability to observe transport performance, and may limit the ability of network operators to trace problems, make appropriate QoS decisions, or response to other queries about the network service. For some this will be blessing, for others it may be a curse. For example, operational performance data about encrypted flows needs to be determined by traffic pattern analysis, rather than relying on traditional tools. This can impact the ability of the operator to respond to faults, it could require reliance on endpoint diagnostic tools or user involvement in diagnosing and troubleshooting unusual use cases or non-trivial problems. A key need here is for tools to provide useful information during network anomalies (e.g., significant reordering, high or intermittent loss). Although many network operators utilise transport information as a part of their operational practice, the network will not break because transport headers are encrypted, and this may require alternative tools may need to be developed and deployed.

2.3.1. Examples of measurements

Measurements can be used to monitor the health of a portion of the Internet, to provide early warning of the need to take action. They can assist in debugging and diagnosing the root causes of faults that concern a particular user's traffic. They can also be used to support post-mortem investigation after an anomaly to determine the root cause of a problem.

In some case, measurements may involve active injection of test traffic to complete a measurement. However, most operators do not have access to user equipment, and injection of test traffic may be associated with costs in running such tests (e.g., the implications of bandwidth tests in a mobile network are obvious). Some active measurements (e.g., response under load or particular workloads) perturb other traffic, and could require dedicated access to the network segment. An alternative approach is to use in-network techniques that observe transport packet headers in operational networks to make the measurements.

In other cases, measurement involves dissecting network traffic flows. The observed transport layer information can help identify whether the link/network tuning is effective and alert to potential problems that can be hard to derive from link or device measurements alone. The design trade-offs for radio networks are often very different to those of wired networks. A radio-based network (e.g., cellular mobile, enterprise WiFi, satellite access/back-haul, point-to-point radio) has the complexity of a subsystem that performs radio resource management, with direct impact on the available capacity, and potentially loss/reordering of packets. The impact of the pattern of loss and congestion, differs for different traffic types, correlation with propagation and interference can all have significant impact on the cost and performance of a provided service. The need for this type of information is expected to increase as operators bring together heterogeneous types of network equipment and seek to deploy opportunistic methods to access radio spectrum.

2.4. Observing Headers to Implement Network Policy

Information from the transport protocol can be used by a multi-field classifier as a part of policy framework. Policies are commonly used for management of the QoS or Quality of Experience (QoE) in resource-constrained networks and by firewalls that use the information to implement access rules (see also section 2.2.2 of [I-D.mm-wg-effect-encrypt]). Traffic that cannot be classified, will typically receive a default treatment.

3. Encryption and Authentication of Transport Headers

End-to-end encryption can be applied at various protocol layers. It can be applied above the transport to encrypt the transport payload. Encryption methods can hide information from an eavesdropper in the network. Encryption can also help protect the privacy of a user, by hiding data relating to user/device identity or location. Neither an integrity check nor encryption methods prevent traffic analysis, and usage needs to reflect that profiling of users, identification of location and fingerprinting of behaviour can take place even on encrypted traffic flows.

There are several motivations:

- o One motive to use encryption is a response to perceptions that the network has become ossified by over-reliance on middleboxes that prevent new protocols and mechanisms from being deployed. This has led to a perception that there is too much "manipulation" of protocol headers within the network, and that designing to deploy in such networks is preventing transport evolution. In the light of this, a method that authenticates transport headers may help improve the pace of transport development, by eliminating the need to always consider deployed middleboxes [I-D.trammell-plus-abstract-mech], or potentially to only explicitly enable middlebox use for particular paths with particular middleboxes that are deliberately deployed to realise a useful function for the network and/or users[RFC3135].
- o Another motivation stems from increased concerns about privacy and surveillance. Some Internet users have valued the ability to protect identity, user location, and defend against traffic analysis, and have used methods such as IPsec Encapsulated Security Payload (ESP), Virtual Private Networks (VPNs) and other encrypted tunnel technologies. Revelations about the use of pervasive surveillance [RFC7624] have, to some extent, eroded trust in the service offered by network operators, and following the Snowden revelation in the USA in 2013 has led to an increased desire for people to employ encryption to avoid unwanted "eavesdropping" on their communications. Concerns have also been voiced about the addition of information to packets by third parties to provide analytics, customization, advertising, cross-site tracking of users, to bill the customer, or to selectively allow or block content. Whatever the reasons, there are now activities in the IETF to design new protocols that may include some form of transport header encryption (e.g., QUIC [I-D.ietf-quic-transport]).

Authentication methods (that provide integrity checks of protocols fields) have also been specified at the network layer, and this also protects transport header fields. The network layer itself carries protocol header fields that are increasingly used to help forwarding decisions reflect the need of transport protocols, such as the IPv6 Flow Label [RFC6437], the DSCP and ECN.

The use of transport layer authentication and encryption exposes a tussle between middlebox vendors, operators, applications developers and users.

- o On the one hand, future Internet protocols that enable large-scale encryption assist in the restoration of the end-to-end nature of the Internet by returning complex processing to the endpoints, since middleboxes cannot modify what they cannot see.
- o On the other hand, encryption of transport layer header information has implications for people who are responsible for operating networks and researchers and analysts seeking to

understand the dynamics of protocols and traffic patterns.

Whatever the motives, a decision to use pervasive of transport header encryption will have implications on the way in which design and evaluation is performed, and which can in turn impact the direction of evolution of the TCP/IP stack. While the IETF can specify protocols, the success in actual deployment is often determined by many factors [RFC5218] that are not always clear at the time when protocols are being defined.

The next subsections briefly review some security design options for transport protocols. A Survey of Transport Security Protocols [I-D.ietf-taps-transport-security] provides more details concerning commonly used encryption methods at the transport layer.

3.1. Authenticating the Transport Protocol Header

Transport layer header information can be authenticated. An integrity check that protects the immutable transport header fields, but can still expose the transport protocol header information in the clear, allowing in-network devices to observe these fields. An integrity check can not prevent in-network modification, but can avoid a receiver accepting changes and avoid impact on the transport protocol operation.

An example transport authentication mechanism is TCP-Authentication (TCP-AO) [RFC5925]. This TCP option authenticates the IP pseudo header, TCP header, and TCP data. TCP-AO protects the transport layer, preventing attacks from disabling the TCP connection itself and provides replay protection. TCP-AO may interact with middleboxes, depending on their behaviour [RFC3234].

The IPsec Authentication Header (AH) [RFC4302] was designed to work at the network layer and authenticate the IP payload. This approach authenticates all transport headers, and verifies their integrity at the receiver, preventing in-network modification.

3.2. Encrypting the Transport Payload

The transport layer payload can be encrypted to protect the content of transport segments. This leaves transport protocol header information in the clear. The integrity of immutable transport header fields could be protected by combining this with an integrity check (Section 3.1).

Examples of encrypting the payload include Transport Layer Security (TLS) over TCP [RFC5246] [RFC7525], Datagram TLS (DTLS) over UDP [RFC6347] [RFC7525], and TCPCrypt [I-D.ietf-tcpinc-tcpencrypt], which permits opportunistic encryption of the TCP transport payload.

3.3. Encrypting the Transport Header

The network layer payload could be encrypted (including the entire transport header and the payload). This method provides confidentiality of the entire transport packet. It therefore does not expose any transport information to devices in the network, which also prevents modification along a network path.

One example of encryption at the network layer is use of IPsec Encapsulating Security Payload (ESP) [RFC4303] in tunnel mode. This encrypts and authenticates all transport headers, preventing visibility of the transport headers by in-network devices. Some Virtual Private Network (VPN) methods also encrypt these headers.

3.4. Authenticating Transport Information and Selectively Encrypting the Transport Header

A transport protocol design can encrypt selected header fields, while also choosing to authenticate fields in the transport header. This allows specific transport header fields to be made observable by network devices. End-to end integrity checks can prevent an endpoint from undetected modification of the immutable transport headers.

Mutable fields in the transport header provide opportunities for middleboxes to modify the transport behaviour (e.g., the extended headers described in [I-D.trammell-plus-abstract-mech]). This considers only immutable fields in the transport headers, that is, fields that may be authenticated End-to-End across a path.

An example of a method that encrypts some, but not all, transport information is GRE-in-UDP [RFC8086] when used with GRE encryption.

3.5. Optional Encryption of Header Information

There are implications to the use of optional header encryption in the design of a transport protocol, where support of optional mechanisms can increase the complexity of the protocol and its implementation and in the management decisions that are required to use variable format fields. Instead, fields of a specific type ought to always be sent with the same level of confidentiality or integrity protection.

4. Addition of Transport Information to Network-Layer Protocol Headers

Transport protocol information can be made visible in a network-layer header. This has the advantage that this information can then be observed by in-network devices. This has the advantage that a single header can support all transport protocols, but there may also be less desirable implications of separating the operation of the transport protocol from the measurement framework.

Some measurements may be made by adding additional protocol headers carrying operations, administration and management (OAM) information to packets at the ingress to a maintenance domain (e.g., an Ethernet protocol header with timestamps and sequence number information using a method such as 802.1lag or in-situ OAM [I-D.ietf-ippm-ioam-data]) and removing the additional header at the egress of the maintenance domain. This approach enables some types of measurements, but does not cover the entire range of measurements described in this document. In some cases, it can be difficult to position measurement tools at the required segments/nodes and there can be challenges in correlating the downstream/upstream information when in-band OAM data is inserted by an on-path device.

Another example of a network-layer approach is the IPv6 Performance and Diagnostic Metrics (PDM) Destination Option [I-D.ietf-ippm-6man-pdm-option]. This allows a sender to optionally include a destination option that carries header fields that can be used to observe timestamps and packet sequence numbers. This information could be authenticated by receiving transport endpoints when the information is added at the sender and visible at the receiving endpoint, although methods to do this have not currently been proposed. This method needs to be explicitly enabled at the sender.

It can be undesirable to rely on methods requiring the presence of network options or extension headers. IPv4 network options are often not supported (or are carried on a slower processing path) and some IPv6 networks are also known to drop packets that set an IPv6 header extension (e.g., [RFC7872]). Another disadvantage is that protocols that separately expose header information do not necessarily have an advantage to expose the information that is utilised by the protocol itself, and could manipulate this header information to gain an advantage from the network.

5. Implications of Protecting the Transport Headers

The choice of which fields to expose and which to encrypt is a design choice for the transport protocol. Any selective encryption method requires trading two conflicting goals for a transport protocol designer to decide which header fields to encrypt. Security work typically employs a design technique that seeks to expose only what is needed. However, there can be performance and operational benefits in exposing selected information to network tools.

This section explores key implications of working with encrypted transport protocols.

5.1. Independent Measurement

Independent observation by multiple actors is important for scientific analysis. Encrypting transport header encryption changes the ability for other actors to collect and independently analyse data. Internet transport protocols employ a set of mechanisms. Some of these need to work in cooperation with the network layer - loss detection and recovery, congestion detection and congestion control, some of these need to work only End-to-End (e.g., parameter negotiation, flow-control).

When encryption conceals information in the transport header, it could be possible for an applications to provide summary data on performance and usage of the network. This data could be made available to other actors. However, this data needs to contain sufficient detail to understand (and possibly reconstruct the network traffic pattern for further testing) and to be correlated with the configuration of the network paths being measured.

Sharing information between actors needs also to consider the privacy of the user and the incentives for providing accurate and detailed information. Protocols that expose the state information used by the transport protocol in their header information (e.g., timestamps used to calculate the RTT, packet numbers used to assess congestion and requests for retransmission) provide an incentive for the sending endpoint to provide correct information, increasing confidence that the observer understands the transport interaction with the network. This becomes important when considering changes to transport protocols, changes in network infrastructure, or the emergence of new traffic patterns.

5.2. Characterising "Unknown" Network Traffic

The patterns and types of traffic that share Internet capacity changes with time as networked applications, usage patterns and protocols continue to evolve.

If "unknown" or "uncharacterised" traffic patterns form a small part of the traffic aggregate passing through a network device or segment of the network the path, the dynamics of the uncharacterised traffic may not have a significant collateral impact on the performance of other traffic that shares this network segment. Once the proportion of this traffic increases, the need to monitor the traffic and determine if appropriate safety measures need to be put in place.

Tracking the impact of new mechanisms and protocols requires traffic volume to be measured and new transport behaviours to be identified. This is especially true of protocols operating over a UDP substrate.

The level and style of encryption needs to be considered in determining how this activity is performed. On a shorter timescale, information may also need to be collected to manage denial of service attacks against the infrastructure.

5.3. Accountability and Internet Transport Protocols

Information provided by tools observing transport headers can be used to classify traffic, and to limit the network capacity used by certain flows. Operators can potentially use this information to prioritise or de-prioritise certain flows or classes of flow, with potential implications for network neutrality, or to rate limit malicious or otherwise undesirable flows (e.g., for Distributed Denial of Service, DDOS, protection, or to ensure compliance with a traffic profile Section 2.2.4). Equally, operators could use analysis of transport headers and transport flow state to demonstrate that they are not providing differential treatment to certain flows. Obfuscating or hiding this information using encryption is expected to lead operators and maintainers of middleboxes (firewalls, etc.) to seek other methods to classify, and potentially other mechanisms to condition, network traffic.

A lack of data reduces the level of precision with which flows can be classified and conditioning mechanisms are applied (e.g., rate limiting, circuit breaker techniques [RFC8084], or blocking of uncharacterised traffic), and this needs to be considered when evaluating the impact of designs for transport encryption [RFC5218].

5.4. Impact on Research, Development and Deployment

The majority of present Internet applications use two well-known transport protocols: e.g., TCP and UDP. Although TCP represents the majority of current traffic, some important real-time applications use UDP, and much of this traffic utilises RTP format headers in the payload of the UDP datagram. Since these protocol headers have been fixed for decades, a range of tools and analysis methods have become common and well-understood. Over this period, the transport protocol

headers have mostly changed slowly, and so also the need to develop tools track new versions of the protocol.

Looking ahead, there will be a need to update these protocols and to develop and deploy new transport mechanisms and protocols. There are both opportunities and also challenges to the design, evaluation and deployment of new transport protocol mechanisms.

Integrity checks can protect an endpoint from undetected modification of protocol fields by network devices, whereas encryption and obfuscation can further prevent these headers being utilised by network devices. Hiding headers can therefore provide the opportunity for greater freedom to update the protocols and can ease experimentation with new techniques and their final deployment in endpoints.

Hiding headers can limit the ability to measure and characterise traffic. Measurement data is increasingly being used to inform design decisions in networking research, during development of new mechanisms and protocols and in standardisation. Measurement has a critical role in the design of transport protocol mechanisms and their acceptance by the wider community (e.g., as a method to judge the safety for Internet deployment). Observation of pathologies are also important in understanding the interactions between cooperating protocols and network mechanism, the implications of sharing capacity with other traffic and the impact of different patterns of usage.

Evolution and the ability to understand (measure) the impact need to proceed hand-in-hand. Attention needs to be paid to the expected scale of deployment of new protocols and protocol mechanisms. Whatever the mechanism, experience has shown that it is often difficult to correctly implement combination of mechanisms [RFC8085]. These mechanisms therefore typically evolve as a protocol matures, or in response to changes in network conditions, changes in network traffic or changes to application usage.

New transport protocol formats are expected to facilitate an increased pace of transport evolution, and with it the possibility to experiment with and deploy a wide range of protocol mechanisms. There has been recent interest in a wide range of new transport methods, e.g., Larger Initial Window, Proportional Rate Reduction (PRR), congestion control methods based on measuring bottleneck bandwidth and round-trip propagation time, the introduction of AQM techniques and new forms of ECN response (e.g., Data Centre TCP, DCTP, and methods proposed for L4S). The growth and diversity of applications and protocols using the Internet also continues to expand. For each new method or application it is desirable to build a body of data reflecting its behaviour under a wide range of deployment scenarios, traffic load, and interactions with other deployed/candidate methods.

Open standards motivate a desire for this evaluation to include independent observation and evaluation of performance data, which in turn suggests control over where and when measurement samples are collected. This requires consideration of the appropriate balance between encrypting all and no transport information.

6. Conclusions

The majority of present Internet applications use two well-known transport protocols: e.g., TCP and UDP. Although TCP represents the majority of current traffic, some important real-time applications have used UDP, and much of this traffic utilises RTP format headers in the payload of the UDP datagram. Since these protocol headers have been fixed for decades, a range of tools and analysis methods have become common and well-understood. Over this period, the transport protocol headers have mostly changed slowly, and so also the need to develop tools track new versions of the protocol.

Confidentiality and strong integrity checks have properties that are being incorporated into new protocols and which have important benefits. The pace of development of transports using the WebRTC data channel and the rapid deployment of QUIC prototype transports can both be attributed to using a combination of UDP transport and confidentiality of the UDP payload.

The traffic that can be observed by on-path network devices is a function of transport protocol design/options, network use, applications and user characteristics. In general, when only a small proportion of the traffic has a specific (different) characteristic. Such traffic seldom leads to an operational issue although the ability to measure and monitor it is less. The desire to understand the traffic and protocol interactions typically grows as the proportion of traffic increases in volume. The challenges increase when multiple instances of an evolving protocol contribute to the traffic that share network capacity.

An increased pace of evolution therefore needs to be accompanied by methods that can be successfully deployed and used across operational networks. This leads to a need for network operators (at various level (ISPs, enterprises, firewall maintainer, etc) to identify appropriate operational support functions and procedures.

Protocols that change their transport header format (wire format) or their behaviour (e.g., algorithms that are needed to classify and characterise the protocol), will require new tooling needs to be developed to catch-up with the changes. If the currently deployed tools and methods are no longer relevant and performance may not be correctly measured. This can increase the response-time after faults, and can impact the ability to manage the network resulting in traffic causing traffic to be treated inappropriately (e.g., rate limiting because of being incorrectly classified/monitored). There are benefits in exposing consistent information to the network that avoids traffic being mis-classified and then receiving a default treatment by the network.

As a part of its design a new protocol specification therefore needs to weigh the benefits of ossifying common headers, versus the potential demerits of exposing specific information that could be observed along the network path to provide tools to manage new variants of protocols. Several scenarios to illustrate different ways this could evolve are provided below:

- o One scenario is when transport protocols provide consistent information to the network by intentionally exposing a part of the transport header. The design fixes the format of this information between versions of the protocol. This ossification of the transport header allows an operator to establish tooling and procedures that enable it to provide consistent traffic management as the protocol evolves. In contrast to TCP (where all protocol information is exposed), evolution of the transport is facilitated by providing cryptographic integrity checks of the transport header fields (preventing undetected middlebox changes) and encryption of other protocol information (preventing observation within the network, or incentivising the use of the exposed information, rather than inferring information from other characteristics of the flow traffic). The exposed transport information can be used by operators to provide troubleshooting, measurement and any necessary functions appropriate to the class of traffic (priority, retransmission, reordering, circuit breakers, etc).

- o An alternative scenario adopts different design goals, with a different outcome. A protocol that encrypts all header information forces network operators to act independently from apps/transport developments to provide the transport information they need. A range of approaches may proliferate, as in current networks, operators can add a shim header to each packet as a flow as it crosses the network; other operators/managers could develop heuristics and pattern recognition to derive information that classifies flows and estimates quality metrics for the service being used; some could decide to rate-limit or block traffic until new tooling is in place. In many cases, the derived information can be used by operators to provide necessary functions appropriate to the class of traffic (priority, retransmission, reordering, circuit breakers, etc). Troubleshooting, and measurement becomes more difficult, and more diverse. This could require additional information beyond that visible in the packet header and when this information is used to inform decisions by on-path devices it can lead to dependency on other characteristics of the flow. In some cases, operators might need access to keying information to interpret encrypted data that they observe. Some use cases could demand use of transports that do not use encryption.

The outcome could have significant implications on the way the Internet architecture develops. It exposes a risk that significant actors (e.g., developers and transport designers) achieve more control of the way in which the Internet architecture develops. In particular, there is a possibility that designs could evolve to significantly benefit of customers for a specific vendor, and that communities with very different network, applications or platforms could then suffer at the expense of benefits to their vendors own customer base. In such a scenario, there could be no incentive to support other applications/products or to work in other networks leading to reduced access for new approaches.

7. Acknowledgements

The authors would like to thank all who have talked to him face-to-face or via email. ...

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421. The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

8. Security Considerations

This document is about design and deployment considerations for transport protocols. Issues relating to security are discussed in the various sections of the document.

Authentication, confidentiality protection, and integrity protection are identified as Transport Features by [RFC8095]. As currently deployed in the Internet, these features are generally provided by a protocol or layer on top of the transport protocol [I-D.ietf-taps-transport-security].

Confidentiality and strong integrity checks have properties that can also be incorporated into the design of a transport protocol. Integrity checks can protect an endpoint from undetected modification of protocol fields by network devices, whereas encryption and obfuscation can further prevent these headers being utilised by network devices. Hiding headers can therefore provide the opportunity for greater freedom to update the protocols and can ease experimentation with new techniques and their final deployment in endpoints. A protocol specification needs to weigh the benefits of ossifying common headers, versus the potential demerits of exposing specific information that could be observed along the network path to provide tools to manage new variants of protocols.

A protocol design that uses header encryption can provide confidentiality of some or all of the protocol header information. This prevents an on-path device from knowledge of the header field. It therefore prevents mechanisms being built that directly rely on the information or seeks to imply semantics of an exposed header field. Hiding headers can limit the ability to measure and characterise traffic.

Exposed transport headers are sometimes utilised as a part of the information to detect anomalies in network traffic. This can be used as the first line of defence to identify potential threats from DOS or malware and redirect suspect traffic to dedicated nodes responsible for DOS analysis, malware detection, or to perform packet scrubbing "Scrubbing" (the normalization of packets so that there are no ambiguities in interpretation by the ultimate destination of the packet). These techniques are currently used by some operators to also defend from distributed DOS attacks.

Exposed transport headers are sometimes also utilised as a part of the information used by the receiver of a transport protocol to protect the transport layer from data injection by an attacker. In evaluating this use of exposed header information, it is important to consider whether it introduces a significant DOS threat. For example, an attacker could construct a DOS attack by sending packets with a sequence number that falls within the currently accepted range of sequence numbers at the receiving endpoint, this would then introduce additional work at the receiving endpoint, even though the data in the attacking packet may not finally be delivered by the transport layer. This is sometimes known as a "shadowing attack". An attack can, for example, disrupt receiver processing, trigger loss and retransmission, or make a receiving endpoint perform unproductive decryption of packets that cannot be successfully decrypted (forcing a receiver to commit decryption resources, or to update and then restore protocol state).

One mitigation to off-path attack is to deny knowledge of what header information is accepted by a receiver or obfuscate the accepted header information, e.g., setting a non-predictable initial value for a sequence number during a protocol handshake, as in [RFC3550] and [RFC6056], or a port value that can not be predicted (see section 5.1 of [RFC8085]). A receiver could also require additional information to be used as a part of check before accepting packets at the transport layer (e.g., utilising a part of the sequence number space that is encrypted; or by verifying an encrypted token not visible to an attacker). This would also mitigate on-path attacks. An additional processing cost can be incurred when decryption needs to be attempted before a receiver is able to discard injected packets.

Open standards motivate a desire for this evaluation to include independent observation and evaluation of performance data, which in turn suggests control over where and when measurement samples are collected. This requires consideration of the appropriate balance between encrypting all and no transport information. Open data, and accessibility to tools that can help understand trends in application deployment, network traffic and usage patterns can all contribute to understanding security challenges.

9. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

10. References

10.1. Normative References

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

10.2. Informative References

[I-D.dolson-plus-middlebox-benefits]
Dolson, D., Snellman, J., Boucadair, M. and C. Jacquenet,
"Beneficial Functions of Middleboxes", Internet-Draft
draft-dolson-plus-middlebox-benefits-03, March 2017.

[I-D.ietf-aqm-codel]
Nichols, K., Jacobson, V., McGregor, A. and J. Iyengar,
"Controlled Delay Active Queue Management", Internet-Draft
draft-ietf-aqm-codel-10, October 2017.

[I-D.ietf-aqm-fq-codel]

Hoeiland-Joergensen, T., McKenney, P., dave.taht@gmail.com, d., Gettys, J. and E. Dumazet, "The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm", Internet-Draft draft-ietf-aqm-fq-codel-06, March 2016.

[I-D.ietf-aqm-pie]

Pan, R., Natarajan, P., Baker, F. and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", Internet-Draft draft-ietf-aqm-pie-10, September 2016.

[I-D.ietf-ippm-6man-pdm-option]

Elkins, N., Hamilton, R. and m. mackermann@bcbsm.com, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", Internet-Draft draft-ietf-ippm-6man-pdm-option-13, June 2017.

[I-D.ietf-ippm-ioam-data]

Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d. and J. Lemon, "Data Fields for In-situ OAM", Internet-Draft draft-ietf-ippm-ioam-data-02, March 2018.

[I-D.ietf-quick-transport]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Internet-Draft draft-ietf-quick-transport-03, May 2017.

[I-D.ietf-taps-transport-security]

Pauly, T., Perkins, C., Rose, K. and C. Wood, "A Survey of Transport Security Protocols", Internet-Draft draft-ietf-taps-transport-security-01, May 2018.

[I-D.ietf-tcpinc-tcpcrypt]

Bittau, A., Giffin, D., Handley, M., Mazieres, D., Slack, Q. and E. Smith, "Cryptographic protection of TCP Streams (tcpcrypt)", Internet-Draft draft-ietf-tcpinc-tcpcrypt-11, November 2017.

[I-D.ietf-tsvwg-l4s-arch]

Briscoe, B., Schepper, K. and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", Internet-Draft draft-ietf-tsvwg-l4s-arch-01, October 2017.

[I-D.mm-wg-effect-encrypt]

Moriarty, K. and A. Morton, "Effects of Pervasive Encryption on Operators", Internet-Draft draft-mm-wg-effect-encrypt-24, March 2018.

[I-D.thomson-quick-grease]

Thomson, M., "More Apparent Randomization for QUIC", Internet-Draft draft-thomson-quic-grease-00, December 2017.

- [I-D.trammell-plus-abstract-mech]
Trammell, B., "Abstract Mechanisms for a Cooperative Path Layer under Endpoint Control", Internet-Draft draft-trammell-plus-abstract-mech-00, September 2016.
- [I-D.trammell-plus-statefulness]
Kuehlewind, M., Trammell, B. and J. Hildebrand, "Transport-Independent Path Layer State Management", Internet-Draft draft-trammell-plus-statefulness-02, December 2016.
- [Latency] Briscoe, B., "Reducing Internet Latency: A Survey of Techniques and Their Merits", November 2014.
- [Measure] Fairhurst, G., Kuehlewind, M. and D. Lopez, "Measurement-based Protocol Design", June 2017.
- [RFC1273] Schwartz, M.F., "Measurement Study of Changes in Service-Level Reachability in the Global TCP/IP Internet: Goals, Experimental Design, Implementation, and Policy Considerations", RFC 1273, DOI 10.17487/RFC1273, November 1991, <<https://www.rfc-editor.org/info/rfc1273>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<http://www.rfc-editor.org/info/rfc2474>>.
- [RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <<https://www.rfc-editor.org/info/rfc2914>>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G. and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<http://www.rfc-editor.org/info/rfc3135>>.
- [RFC3168] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<http://www.rfc-editor.org/info/rfc3168>>.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <<http://www.rfc-editor.org/info/rfc3234>>.

- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G. and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", BCP 69, RFC 3449, DOI 10.17487/RFC3449, December 2002, <<http://www.rfc-editor.org/info/rfc3449>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C. and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<http://www.rfc-editor.org/info/rfc4585>>.
- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S. and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<http://www.rfc-editor.org/info/rfc4737>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC5236] Jayasumana, A., Piratla, N., Banka, T., Bare, A. and R. Whitner, "Improved Packet Reordering Metrics", RFC 5236, DOI 10.17487/RFC5236, June 2008, <<http://www.rfc-editor.org/info/rfc5236>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<http://www.rfc-editor.org/info/rfc5246>>.

- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [RFC5559] Eardley, P., Ed., "Pre-Congestion Notification (PCN) Architecture", RFC 5559, DOI 10.17487/RFC5559, June 2009, <<https://www.rfc-editor.org/info/rfc5559>>.
- [RFC5925] Touch, J., Mankin, A. and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<http://www.rfc-editor.org/info/rfc5925>>.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <<https://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, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<http://www.rfc-editor.org/info/rfc6347>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S. and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<http://www.rfc-editor.org/info/rfc6437>>.
- [RFC6679] Westerlund, M., Johansson, I., Perkins, C., O'Hanlon, P. and K. Carlberg, "Explicit Congestion Notification (ECN) for RTP over UDP", RFC 6679, DOI 10.17487/RFC6679, August 2012, <<http://www.rfc-editor.org/info/rfc6679>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7525] Sheffer, Y., Holz, R. and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<http://www.rfc-editor.org/info/rfc7525>>.
- [RFC7567] Baker, F. Ed., and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<http://www.rfc-editor.org/info/rfc7567>>.

- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C. and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<http://www.rfc-editor.org/info/rfc7624>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T. and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC7928] Kuhn, N., Ed., Natarajan, P., Ed., Khademi, N. Ed., and D. Ros, "Characterization Guidelines for Active Queue Management (AQM)", RFC 7928, DOI 10.17487/RFC7928, July 2016, <<http://www.rfc-editor.org/info/rfc7928>>.
- [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <<http://www.rfc-editor.org/info/rfc8084>>.
- [RFC8085] Eggert, L., Fairhurst, G. and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<http://www.rfc-editor.org/info/rfc8085>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X. and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<http://www.rfc-editor.org/info/rfc8086>>.
- [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <<http://www.rfc-editor.org/info/rfc8087>>.
- [RFC8095] Fairhurst, G., Ed., Trammell, B. Ed., and M. Kuehlewind, Ed., "Services Provided by IETF Transport Protocols and Congestion Control Mechanisms", RFC 8095, DOI 10.17487/RFC8095, March 2017, <<https://www.rfc-editor.org/info/rfc8095>>.
- [RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L. and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <<https://www.rfc-editor.org/info/rfc8257>>.
- [Tor] The Tor Project, ., "<<https://www.torproject.org>>", June 2017.

Appendix A. Revision information

-00 This is an individual draft for the IETF community.

-01 This draft was a result of walking away from the text for a few days and then reorganising the content.

-02 This draft fixes textual errors.

-03 This draft follows feedback from people reading this draft.

-04 This adds an additional contributor and includes significant reworking to ready this for review by the wider IETF community Colin Perkins joined the author list.

Comments from the community are welcome on the text and recommendations.

-05 Corrections received and helpful inputs from Mohamed Boucadair.

-06 Updated following comments from Stephen Farrell, and feedback via email. Added a draft conclusion section to sketch some strawman scenarios that could emerge.

-07 Updated following comments from Al Morton, Chris Seal, and other feedback via email.

-08 Updated to address comments sent to the TSVWG mailing list by Kathleen Moriarty (on 08/05/2018 and 17/05/2018), Joe Touch on 11/05/2018, and Spencer Dawkins.

-09 Updated security considerations.

Authors' Addresses

Godred Fairhurst
University of Aberdeen
Department of Engineering
Fraser Noble Building
Aberdeen, AB24 3UE
Scotland

Email: gorry@erg.abdn.ac.uk
URI: <http://www.erg.abdn.ac.uk/>

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow, G12 8QQ
Scotland

Email: csp@csp Perkins.org
URI: <https://csp Perkins.org/>

TSVWG
Internet-Draft
Intended status: Informational
Expires: February 28, 2019

G. Fairhurst
University of Aberdeen
C. Perkins
University of Glasgow
August 27, 2018

The Impact of Transport Header Confidentiality on Network Operation and
Evolution of the Internet
draft-fairhurst-tsvwg-transport-encrypt-10

Abstract

This document describes implications of applying end-to-end encryption at the transport layer. It identifies in-network uses of transport layer header information. It then reviews the implications of developing end-to-end transport protocols that use authentication to protect the integrity of transport information or encryption to provide confidentiality of the transport protocol header and expected implications of transport protocol design and network operation. Since transport measurement and analysis of the impact of network characteristics have been important to the design of current transport protocols, it also considers the impact on transport and application evolution.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 28, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Context and Rationale	3
3. Current uses of Transport Headers within the Network	9
3.1. Observing Transport Information in the Network	9
3.2. Transport Measurement	15
3.3. Use for Network Diagnostics and Troubleshooting	18
3.4. Observing Headers to Implement Network Policy	19
4. Encryption and Authentication of Transport Headers	19
4.1. Authenticating the Transport Protocol Header	21
4.2. Encrypting the Transport Payload	22
4.3. Encrypting the Transport Header	22
4.4. Authenticating Transport Information and Selectively Encrypting the Transport Header	22
4.5. Optional Encryption of Header Information	23
5. Addition of Transport Information to Network-Layer Protocol Headers	23
6. Implications of Protecting the Transport Headers	24
6.1. Independent Measurement	24
6.2. Characterising "Unknown" Network Traffic	25
6.3. Accountability and Internet Transport Protocols	25
6.4. Impact on Research, Development and Deployment	26
7. Conclusions	27
8. Security Considerations	29
9. IANA Considerations	31
10. Acknowledgements	31
11. Informative References	31
Appendix A. Revision information	37
Authors' Addresses	37

1. Introduction

This document describes implications of applying end-to-end encryption at the transport layer. It reviews the implications of developing end-to-end transport protocols that use encryption to provide confidentiality of the transport protocol header and expected implications of transport protocol design and network operation. It

also considers anticipated implications on transport and application evolution.

2. Context and Rationale

The transport layer provides end-to-end interactions between endpoints (processes) using an Internet path. Transport protocols layer directly over the network-layer service and are sent in the payload of network-layer packets. They support end-to-end communication between applications, supported by higher-layer protocols, running on the end systems (or transport endpoints). This simple architectural view hides one of the core functions of the transport, however, to discover and adapt to the properties of the Internet path that is currently being used. The design of Internet transport protocols is as much about trying to avoid the unwanted side effects of congestion on a flow and other capacity-sharing flows, avoiding congestion collapse, adapting to changes in the path characteristics, etc., as it is about end-to-end feature negotiation, flow control and optimising for performance of a specific application.

To achieve stable Internet operations the IETF transport community has to date relied heavily on measurement and insights of the network operations community to understand the trade-offs, and to inform selection of appropriate mechanisms, to ensure a safe, reliable, and robust Internet (e.g., [RFC1273]). In turn, the network operations community relies on being able to understand the pattern and requirements of traffic passing over the Internet, both in aggregate and at the flow level.

There are many motivations for deploying encrypted transports [RFC7624] (i.e., transport protocols that use encryption to provide confidentiality of some or all of the transport-layer header information), and encryption of transport payloads (i.e. confidentiality of the payload data). The increasing public concerns about the interference with Internet traffic have led to a rapidly expanding deployment of encryption to protect end-user privacy, in protocols like QUIC [I-D.ietf-quic-transport], but also expected to form a basis of future protocol designs.

Some network operators and access providers, have come to rely on the in-network measurement of transport properties and the functionality provided by middleboxes to both support network operations and enhance performance. There can therefore be implications when working with encrypted transport protocols that hide transport header information from the network. These present architectural challenges and considerations in the way transport protocols are designed, and ability to characterise and compare different transport solutions

[Measure], Section 3.2. Implementations of network devices are encouraged to avoid side-effects when protocols are updated. Introducing cryptographic integrity checks to header fields can also prevent undetected manipulation of the field by network devices, or undetected addition of information to a packet. However, this does not prevent inspection of the information by a device on path, and it is possible that such devices could develop mechanisms that rely on the presence of such a field, or a known value in the field.

Reliance on the presence and semantics of specific header information leads to ossification: An endpoint could be required to supply a specific header to receive the network service that it desires. In some cases, this could be benign or advantageous to the protocol (e.g., recognising the start of a connection, or explicitly exposing protocol information can be expected to provide more consistent decisions by on-path devices than the use of diverse methods to infer semantics from other flow properties). In some cases, this is not beneficial (e.g., a mechanism implemented in a network device, such as a firewall, that required a header field to have only a specific known set of values could prevent the device from forwarding packets using a different version of a protocol that introduces a new feature that changes the value present in this field, preventing evolution of the protocol).

Examples of the impact of ossification on transport protocol design and ease of deployment can be seen in the case of Multipath TCP (MPTCP) and the TCP Fast Open option. The design of MPTCP had to be revised to account for middleboxes, so called "TCP Normalizers", that monitor the evolution of the window advertised in the TCP headers and that reset connections if the window does not grow as expected. Similarly, TCP Fast Open has had issues with middleboxes that remove unknown TCP options, that drop segments with unknown TCP options, that drop segments that contain data and have the SYN bit set, that drop packets with SYN/ACK that acknowledge data, or that disrupt connections that send data before the three-way handshake completes. In both cases, the issue was caused by middleboxes that had a hard-coded understanding of transport behaviour, and that interacted poorly with transports that tried to change that behaviour. Other examples have included middleboxes that rewrite TCP sequence and acknowledgement numbers but are unaware of the (newer) SACK option and don't correctly rewrite selective acknowledgements to match the changes made to the fixed TCP header; or devices that inspect, and change, TCP MSS options that can interfere with path MTU discovery.

A protocol design that uses header encryption can provide confidentiality of some or all of the protocol header information. This prevents an on-path device from knowledge of the header field. It therefore prevents mechanisms being built that directly rely on

the information or seeks to imply semantics of an exposed header field. Using encryption to provide confidentiality of the transport layer brings some well-known privacy and security benefits and can therefore help reduce ossification of the transport layer. In particular, it is important that protocols either do not expose information where the usage may change in future protocols, or that methods that utilise the information are robust to potential changes as protocols evolve over time. To avoid unwanted inspection, a protocol could also intentionally vary the format and value of header fields (sometimes known as Greasing [I-D.thomson-quic-grease]). However, while encryption hides the protocol header information, it does not prevent ossification of the network service: People seeking understanding of network traffic could come to rely on pattern inferences and other heuristics as the basis for network decision and to derive measurement data, creating new dependencies on the transport protocol.

A level of ossification of the transport header can offer trade-offs around authentication, and confidentiality of transport protocol headers and has the potential to explicitly support for other uses of this header information. For example, a design that provides confidentiality of protocol header information can impact the following activities that rely on measurement and analysis of traffic flows:

Network Operations and Research: Observable transport headers enable both operators and the research community to measure and analyse protocol performance, network anomalies, and failure pathologies.

This information can help inform capacity planning, and assist in determining the need for equipment and/or configuration changes by network operators.

The data can also inform Internet engineering research, and help in the development of new protocols, methodologies, and procedures. Concealing the transport protocol header information makes the stream performance unavailable to passive observers along the path, and likely leads to the development of alternative methods to collect or infer that data.

Providing confidentiality of the transport payload, but leaving some, or all, of the transport headers unencrypted, possibly with authentication, can provide the majority of the privacy and security benefits while allowing some measurement.

Protection from Denial of Service: Observable transport headers currently provide useful input to classify traffic and detect anomalous events (e.g., changes in application behaviour,

distributed denial of service attacks). To be effective, this protection needs to be able to uniquely disambiguate unwanted traffic. An inability to separate this traffic using packet header information may result in less-efficient identification of unwanted traffic or development of different methods (e.g. rate-limiting of uncharacterised traffic).

Network Troubleshooting and Diagnostics: Encrypting transport header information eliminates the incentive for operators to troubleshoot what they cannot interpret. A flow experiencing packet loss or jitter looks like an unaffected flow when only observing network layer headers (if transport sequence numbers and flow identifiers are obscured). This limits understanding of the impact of packet loss or latency on the flows, or even localizing the network segment causing the packet loss or latency. Encrypted traffic may imply "don't touch" to some, and could limit a trouble-shooting response to "can't help, no trouble found". The additional mechanisms that will need to be introduced to help reconstruct transport-level metrics add complexity and operational costs (e.g., in deploying additional functions in equipment or adding traffic overhead).

Network Traffic Analysis: Hiding transport protocol header information can make it harder to determine which transport protocols and features are being used across a network segment and to measure trends in the pattern of usage. This could impact the ability for an operator to anticipate the need for network upgrades and roll-out. It can also impact the on-going traffic engineering activities performed by operators (such as determining which parts of the path contribute delay, jitter or loss). While the impact may, in many cases, be small there are scenarios where operators directly support particular services (e.g., to troubleshoot issues relating to Quality of Service, QoS; the ability to perform fast re-routing of critical traffic, or support to mitigate the characteristics of specific radio links). The more complex the underlying infrastructure the more important this impact.

Open and Verifiable Network Data: Hiding transport protocol header information can reduce the range of actors that can capture useful measurement data. For example, one approach could be to employ an existing transport protocol that reveals little information (e.g., UDP), and perform traditional transport functions at higher layers protecting the confidentiality of transport information. Such a design, limits the information sources available to the Internet community to understand the operation of new transport protocols, so preventing access to the information necessary to inform design

decisions and standardisation of the new protocols and related operational practices.

The cooperating dependence of network, application, and host to provide communication performance on the Internet is uncertain when only endpoints (i.e., at user devices and within service platforms) can observe performance, and performance cannot be independently verified by all parties. The ability of other stakeholders to review code can help develop deeper insight. In the heterogeneous Internet, this helps extend the range of topologies, vendor equipment, and traffic patterns that are evaluated.

Independently captured data is important to help ensure the health of the research and development communities. It can provide input and test scenarios to support development of new transport protocol mechanisms, especially when this analysis can be based on the behaviour experienced in a diversity of deployed networks.

Independently verifiable performance metrics might also be important to demonstrate regulatory compliance in some jurisdictions, and provides an important basis for informing design decisions.

The last point leads us to consider the impact of hiding transport headers in the specification and development of protocols and standards. This has potential impact on:

- o Understanding Feature Interactions: An appropriate vantage point, coupled with timing information about traffic flows, provides a valuable tool for benchmarking equipment, functions, and/or configurations, and to understand complex feature interactions. An inability to observe transport protocol information can limit the ability to diagnose and explore interactions between features at different protocol layers, a side-effect of not allowing a choice of vantage point from which this information is observed.
- o Supporting Common Specifications: Transmission Control Protocol (TCP) is currently the predominant transport protocol used over Internet paths. Its many variants have broadly consistent approaches to avoiding congestion collapse, and to ensuring the stability of the Internet. Increased use of transport layer encryption can overcome ossification, allowing deployment of new transports and different types of congestion control. This flexibility can be beneficial, but it can come at the cost of fragmenting the ecosystem. There is little doubt that developers will try to produce high quality transports for their intended target uses, but it is not clear there are sufficient incentives

to ensure good practice that benefits the wide diversity of requirements for the Internet community as a whole. Increased diversity, and the ability to innovate without public scrutiny, risks point solutions that optimise for specific needs, but accidentally disrupt operations of/in different parts of the network. The social contract that maintains the stability of the Internet relies on accepting common specifications, and on the ability to verify that others also conform.

- o Operational practice: Published transport specifications allow operators to check compliance. This can bring assurance to those operating networks, often avoiding the need to deploy complex techniques that routinely monitor and manage TCP/IP traffic flows (e.g. Avoiding the capital and operational costs of deploying flow rate-limiting and network circuit-breaker methods [RFC8084]). When it is not possible to observe transport header information, methods are still needed to confirm that the traffic produced conforms to the expectations of the operator or developer.
- o Restricting research and development: Hiding transport information can impede independent research into new mechanisms, measurement of behaviour, and development initiatives. Experience shows that transport protocols are complicated to design and complex to deploy, and that individual mechanisms need to be evaluated while considering other mechanisms, across a broad range of network topologies and with attention to the impact on traffic sharing the capacity. If this results in reduced availability of open data, it could eliminate the independent self-checks to the standardisation process that have previously been in place from research and academic contributors (e.g., the role of the IRTF ICCRG, and research publications in reviewing new transport mechanisms and assessing the impact of their experimental deployment)

In summary, there are trade offs. On the one hand, protocol designers have often ignored the implications of whether the information in transport header fields can or will be used by in-network devices, and the implications this places on protocol evolution. This motivates a design that provides confidentiality of the header information. On the other hand, it can be expected that a lack of visibility of transport header information can impact the ways that protocols are deployed, standardised, and their operational support. The choice of whether future transport protocols encrypt their protocol headers therefore needs to be taken based not solely on security and privacy considerations, but also taking into account the impact on operations, standards, and research. Any new Internet transport need to provide appropriate transport mechanisms and operational support to assure the resulting traffic can not result in

persistent congestion collapse [RFC2914]. This document suggests that the balance between information exposed and concealed should be carefully considered when specifying new protocols.

3. Current uses of Transport Headers within the Network

Despite transport headers having end-to-end meaning, some of these transport headers have come to be used in various ways within the Internet. In response to pervasive monitoring [RFC7624] revelations and the IETF consensus that "Pervasive Monitoring is an Attack" [RFC7258], efforts are underway to increase encryption of Internet traffic,. Applying confidentiality to transport header fields would affect how protocol information is used [RFC8404]. To understand these implications, it is first necessary to understand how transport layer headers are currently observed and/or modified by middleboxes within the network.

Transport protocols can be designed to encrypt or authenticate transport header fields. Authentication at the transport layer can be used to detect any changes to an immutable header field that were made by a network device along a path. The intentional modification of transport headers by middleboxes (such as Network Address Translation, NAT, or Firewalls) is not considered. Common issues concerning IP address sharing are described in [RFC6269].

3.1. Observing Transport Information in the Network

If in-network observation of transport protocol headers is needed, this requires knowledge of the format of the transport header:

- o Flows need to be identified at the level required to perform the observation;
- o The protocol and version of the header need to be visible. As protocols evolve over time and there may be a need to introduce new transport headers. This may require interpretation of protocol version information or connection setup information;
- o The location and syntax of any observed transport headers needs to be known. IETF transport protocols can specify this information.

The following subsections describe various ways that observable transport information has been utilised.

3.1.1. Flow Identification

Transport protocol header information (together with information in the network header), has been used to identify a flow and the connection state of the flow, together with the protocol options being used. In some usages, a low-numbered (well-known) transport port number has been used to identify a protocol (although port information alone is not sufficient to guarantee identification of a protocol, since applications can use arbitrary ports, multiple sessions can be multiplexed on a single port, and ports can be re-used by subsequent sessions).

Transport protocols, such as TCP and Stream Control Transport Protocol (SCTP) specify a standard base header that includes sequence number information and other data, with the possibility to negotiate additional headers at connection setup, identified by an option number in the transport header. UDP-based protocols can use, but sometimes do not use, well-known port numbers. Some flows can instead be identified by signalling protocols or through the use of magic numbers placed in the first byte(s) of the datagram payload.

Flow identification is a common function. For example, performed by measurement activities, QoS classification, firewalls, Denial of Service, DOS, prevention. It becomes more complex and less easily achieved when multiplexing is used at or above the transport layer.

3.1.2. Metrics derived from Transport Layer Headers

Some actors manage their portion of the Internet by characterizing the performance of link/network segments. Passive monitoring uses observed traffic to make inferences from transport headers to derive these measurements. A variety of open source and commercial tools have been deployed that utilise this information. The following metrics can be derived from transport header information:

Traffic Rate and Volume: Header information e.g., (sequence number, length) allows derivation of volume measures per-application, to characterise the traffic that uses a network segment or the pattern of network usage. This may be measured per endpoint or for an aggregate of endpoints (e.g., by an operator to assess subscriber usage). It can also be used to trigger measurement-based traffic shaping and to implement QoS support within the network and lower layers. Volume measures can be valuable for capacity planning (providing detail of trends rather than the volume per subscriber).

Loss Rate and Loss Pattern: Flow loss rate may be derived (e.g., from sequence number) and has been used as a metric for

performance assessment and to characterise transport behaviour. Understanding the root cause of loss can help an operator determine whether this requires corrective action. Network operators have used the variation in patterns of loss as a key performance metric, utilising this to detect changes in the offered service.

There are various causes of loss, including: corruption of link frames (e.g., interference on a radio link), buffer overflow (e.g., due to congestion), policing (traffic management), buffer management (e.g., Active Queue Management, AQM [RFC7567]), inadequate provision of traffic preemption. Understanding flow loss rate requires either maintaining per flow packet counters or by observing sequence numbers in transport headers. Loss can be monitored at the interface level by devices in the network. It is often important to understand the conditions under which packet loss occurs. This usually requires relating loss to the traffic flowing on the network node/segment at the time of loss.

Observation of transport feedback information (observing loss reports, e.g., RTP Control Protocol (RTCP) [RFC3550], TCP SACK) can increase understanding of the impact of loss and help identify cases where loss may have been wrongly identified, or the transport did not require the lost packet. It is sometimes more important to understand the pattern of loss, than the loss rate, because losses can often occur as bursts, rather than randomly-timed events.

Throughput and Goodput: The throughput achieved by a flow can be determined even when a flow is encrypted, providing the individual flow can be identified. Goodput [RFC7928] is a measure of useful data exchanged (the ratio of useful/total volume of traffic sent by a flow). This requires ability to differentiate loss and retransmission of packets (e.g., by observing packet sequence numbers in the TCP or the Real Time Protocol, RTP, headers [RFC3550]).

Latency: Latency is a key performance metric that impacts application response time and user-perceived response time. It often indirectly impacts throughput and flow completion time. Latency determines the reaction time of the transport protocol itself, impacting flow setup, congestion control, loss recovery, and other transport mechanisms. The observed latency can have many components [Latency]. Of these, unnecessary/unwanted queuing in network buffers has often been observed as a significant factor. Once the cause of unwanted latency has been identified, this can often be eliminated.

To measure latency across a part of a path, an observation point can measure the experienced round trip time (RTT) using packet sequence numbers, and acknowledgements, or by observing header timestamp information. Such information allows an observation point in the network to determine not only the path RTT, but also to measure the upstream and downstream contribution to the RTT. This has been used to locate a source of latency, e.g., by observing cases where the ratio of median to minimum RTT is large for a part of a path.

The service offered by operators can benefit from latency information to understand the impact of deployment and tune deployed services. Latency metrics are key to evaluating and deploying AQM [RFC7567], DiffServ [RFC2474], and Explicit Congestion Notification (ECN) [RFC3168] [RFC8087]. Measurements could identify excessively large buffers, indicating where to deploy or configure AQM. An AQM method is often deployed in combination with other techniques, such as scheduling [RFC7567] [RFC8290] and although parameter-less methods are desired [RFC7567], current methods [RFC8290] [RFC8289] [RFC8033] often cannot scale across all possible deployment scenarios.

Variation in delay: Some network applications are sensitive to small changes in packet timing. To assess the performance of such applications, it can be necessary to measure the variation in delay observed along a portion of the path [RFC3393] [RFC5481]. The requirements resemble those for the measurement of latency.

Flow Reordering: Significant flow reordering can impact time-critical applications and can be interpreted as loss by reliable transports. Many transport protocol techniques are impacted by reordering (e.g., triggering TCP retransmission, or re-buffering of real-time applications). Packet reordering can occur for many reasons (from equipment design to misconfiguration of forwarding rules). Since this impacts transport performance, network tools are needed to detect and measure unwanted/excessive reordering.

There have been initiatives in the IETF transport area to reduce the impact of reordering within a transport flow, possibly leading to a reduction in the requirements for preserving ordering. These have promise to simplify network equipment design as well as the potential to improve robustness of the transport service. Measurements of reordering can help understand the present level of reordering within deployed infrastructure, and inform decisions about how to progress such mechanisms.

Operational tools to detect mis-ordered packet flows and quantify the degree of reordering. Key performance indicators are retransmission

rate, packet drop rate, sector utilisation level, a measure of reordering, peak rate, the ECN congestion experienced (CE) marking rate, etc.

Metrics have been defined that evaluate whether a network has maintained packet order on a packet-by-packet basis [RFC4737] and [RFC5236].

Techniques for measuring reordering typically observe packet sequence numbers. Some protocols provide in-built monitoring and reporting functions. Transport fields in the RTP header [RFC3550] [RFC4585] can be observed to derive traffic volume measurements and provide information on the progress and quality of a session using RTP. As with other measurement, metadata is often important to understand the context under which the data was collected, including the time, observation point, and way in which metrics were accumulated. The RTCP protocol directly reports some of this information in a form that can be directly visible in the network. A user of summary measurement data needs to trust the source of this data and the method used to generate the summary information.

3.1.3. Metrics derived from Network Layer Headers

Some transport information is made visible in the network-layer protocol header. These header fields are not encrypted and have been utilised to make flow observations.

Use of IPv6 Network-Layer Flow Label: Endpoints are encouraged expose flow information in the IPv6 Flow Label field of the network-layer header (e.g., [RFC8085]). This can be used to inform network-layer queuing, forwarding (e.g., for Equal Cost Multi-Path, ECMP, routing, and Link Aggregation, LAG). This can provide useful information to assign packets to flows in the data collected by measurement campaigns. Although important to characterising a path, it does not directly provide performance data.

Use Network-Layer Differentiated Services Code Point Point:

Applications can expose their delivery expectations to the network by setting the Differentiated Services Code Point (DSCP) field of IPv4 and IPv6 packets. This can be used to inform network-layer queuing and forwarding, and can also provide information on the relative importance of packet information collected by measurement campaigns, but does not directly provide performance data.

This field provides explicit information that can be used in place of inferring traffic requirements (e.g., by inferring QoS requirements from port information via a multi-field classifier).

The DSCP value can therefore impact the quality of experience for a flow. Observations of service performance need to consider this field when a network path has support for differentiated service treatment.

Use of Explicit Congestion Marking: ECN [RFC3168] is an optional transport mechanism that uses a code point in the network-layer header. Use of ECN can offer gains in terms of increased throughput, reduced delay, and other benefits when used over a path that includes equipment that supports an AQM method that performs Congestion Experienced (CE) marking of IP packets [RFC8087].

ECN exposes the presence of congestion on a network path to the transport and network layer. The reception of CE-marked packets can therefore be used to monitor the presence and estimate the level of incipient congestion on the upstream portion of the path from the point of observation (Section 2.5 of [RFC8087]). Because ECN marks are carried in the IP protocol header, it is much easier to measure ECN than to measure packet loss. However, interpreting the marking behaviour (i.e., assessing congestion and diagnosing faults) requires context from the transport layer (path RTT, visibility of loss - that could be due to queue overflow, congestion response, etc) [RFC7567].

Some ECN-capable network devices can provide richer (more frequent and fine-grained) indication of their congestion state. Setting congestion marks proportional to the level of congestion (e.g., Data Center TCP, DCTP [RFC8257], and Low Latency Low Loss Scalable throughput, L4S, [I-D.ietf-tsvwg-l4s-arch]).

Use of ECN requires a transport to feed back reception information on the path towards the data sender. Exposure of this Transport ECN feedback provides an additional powerful tool to understand ECN-enabled AQM-based networks [RFC8087].

AQM and ECN offer a range of algorithms and configuration options, it is therefore important for tools to be available to network operators and researchers to understand the implication of configuration choices and transport behaviour as use of ECN increases and new methods emerge [RFC7567] [RFC8087]. ECN-monitoring is expected to become important as AQM is deployed that supports ECN [RFC8087].

3.2. Transport Measurement

The common language between network operators and application/content providers/users is packet transfer performance at a layer that all can view and analyse. For most packets, this has been transport layer, until the emergence of QUIC, with the obvious exception of Virtual Private Networks (VPNs) and IPsec.

When encryption conceals more layers in each packet, people seeking understanding of the network operation rely more on pattern inferences and other heuristics reliance on pattern inferences and accuracy suffers. For example, the traffic patterns between server and browser are dependent on browser supplier and version, even when the sessions use the same server application (e.g., web e-mail access). It remains to be seen whether more complex inferences can be mastered to produce the same monitoring accuracy (see section 2.1.1 of [RFC8404]).

When measurement datasets are made available by servers or client endpoints, additional metadata, such as the state of the network, is often required to interpret this data. Collecting and coordinating such metadata is more difficult when the observation point is at a different location to the bottleneck/device under evaluation.

Packet sampling techniques can be used to scale the processing involved in observing packets on high rate links. This exports only the packet header information of (randomly) selected packets. The utility of these measurements depends on the type of bearer and number of mechanisms used by network devices. Simple routers are relatively easy to manage, a device with more complexity demands understanding of the choice of many system parameters. This level of complexity exists when several network methods are combined.

This section discusses topics concerning observation of transport flows, with a focus on transport measurement.

3.2.1. Point of Measurement

Often measurements can only be understood in the context of the other flows that share a bottleneck. A simple example is monitoring of AQM. For example, FQ-CODEL [RFC8290], combines sub queues (statistically assigned per flow), management of the queue length (CODEL), flow-scheduling, and a starvation prevention mechanism. Usually such algorithms are designed to be self-tuning, but current methods typically employ heuristics that can result in more loss under certain path conditions (e.g., large RTT, effects of multiple bottlenecks [RFC7567]).

In-network measurements can distinguish between upstream and downstream metrics with respect to a measurement point. These are particularly useful for locating the source of problems or to assess the performance of a network segment or a particular device configuration. By correlating observations of headers at multiple points along the path (e.g., at the ingress and egress of a network segment), an observer can determine the contribution of a portion of the path to an observed metric (to locate a source of delay, jitter, loss, reordering, congestion marking, etc.).

3.2.2. Use by Operators to Plan and Provision Networks

Traffic measurements (e.g., traffic volume, loss, latency) is used by operators to help plan deployment of new equipment and configurations in their networks. Data is also important to equipment vendors who need to understand traffic trends and patterns of usage as inputs to decisions about planning products and provisioning for new deployments. This measurement information can also be correlated with billing information when this is also collected by an operator.

A network operator supporting traffic that uses transport header encryption may not have access to per-flow measurement data. Trends in aggregate traffic can be observed and can be related to the endpoint addresses being used, but it may not be possible to correlate patterns in measurements with changes in transport protocols (e.g., the impact of changes in introducing a new transport protocol mechanism). This increases the dependency on other indirect sources of information to inform planning and provisioning.

3.2.3. Service Performance Measurement

Traffic measurements (e.g., traffic volume, loss, latency) can be used by various actors to help analyse the performance offered to the users of a network segment, and inform operational practice.

While active measurements may be used in-network, passive measurements can have advantages in terms of eliminating unproductive test traffic, reducing the influence of test traffic on the overall traffic mix, and the ability to choose the point of measurement Section 3.2.1. However, passive measurements may rely on observing transport headers.

3.2.4. Measuring Transport to Support Network Operations

Information provided by tools observing transport headers can help determine whether mechanisms are needed in the network to prevent flows from acquiring excessive network capacity. Operators can implement operational practices to manage traffic flows (e.g., to

prevent flows from acquiring excessive network capacity under severe congestion) by deploying rate-limiters, traffic shaping or network transport circuit breakers [RFC8084].

Congestion Control Compliance of Traffic: Congestion control is a key transport function [RFC2914]. Many network operators implicitly accept that TCP traffic to comply with a behaviour that is acceptable for use in the shared Internet. TCP algorithms have been continuously improved over decades, and they have reached a level of efficiency and correctness that custom application-layer mechanisms will struggle to easily duplicate [RFC8085].

A standards-compliant TCP stack provides congestion control may therefore be judged safe for use across the Internet. Applications developed on top of well-designed transports can be expected to appropriately control their network usage, reacting when the network experiences congestion, by back-off and reduce the load placed on the network. This is the normal expected behaviour for IETF-specified transport (e.g., TCP and SCTP).

However, when anomalies are detected, tools can interpret the transport protocol header information to help understand the impact of specific transport protocols (or protocol mechanisms) on the other traffic that shares a network. An observation in the network can gain understanding of the dynamics of a flow and its congestion control behaviour. Analysing observed packet sequence numbers can be used to help build confidence that an application flow backs-off its share of the network load in the face of persistent congestion, and hence to understand whether the behaviour is appropriate for sharing limited network capacity. For example, it is common to visualise plots of TCP sequence numbers versus time for a flow to understand how a flow shares available capacity, deduce its dynamics in response to congestion, etc.

Congestion Control Compliance for UDP traffic UDP provides a minimal message-passing datagram transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and other protocols that choose to use UDP as a transport are required to employ mechanisms to prevent congestion collapse, avoid unacceptable contributions to jitter/latency, and to establish an acceptable share of capacity with concurrent traffic [RFC8085].

A network operator needs tools to understand if datagram flows comply with congestion control expectations and therefore whether there is a need to deploy methods such as rate-limiters, transport

circuit breakers or other methods to enforce acceptable usage for the offered service.

UDP flows that expose a well-known header by specifying the format of header fields can allow information to be observed to gain understanding of the dynamics of a flow and its congestion control behaviour. For example, tools exist to monitor various aspects of the RTP and RTCP header information of real-time flows (see Section 3.1.2).

3.3. Use for Network Diagnostics and Troubleshooting

Transport header information can be useful for a variety of operational tasks [RFC8404]: to diagnose network problems, assess network provider performance, evaluate equipment/protocol performance, capacity planning, management of security threats (including denial of service), and responding to user performance questions. Sections 3.1.2 and 5 of [RFC8404] provide further examples. These tasks seldom involve the need to determine the contents of the transport payload, or other application details.

A network operator supporting traffic that uses transport header encryption can see only encrypted transport headers. This prevents deployment of performance measurement tools that rely on transport protocol information. Choosing to encrypt all the information reduces the operator's ability to observe transport performance, and may limit the ability of network operators to trace problems, make appropriate QoS decisions, or response to other queries about the network service. For some this will be blessing, for others it may be a curse. For example, operational performance data about encrypted flows needs to be determined by traffic pattern analysis, rather than relying on traditional tools. This can impact the ability of the operator to respond to faults, it could require reliance on endpoint diagnostic tools or user involvement in diagnosing and troubleshooting unusual use cases or non-trivial problems. A key need here is for tools to provide useful information during network anomalies (e.g., significant reordering, high or intermittent loss). Although many network operators utilise transport information as a part of their operational practice, the network will not break because transport headers are encrypted, and this may require alternative tools may need to be developed and deployed.

3.3.1. Examples of measurements

Measurements can be used to monitor the health of a portion of the Internet, to provide early warning of the need to take action. They can assist in debugging and diagnosing the root causes of faults that

concern a particular user's traffic. They can also be used to support post-mortem investigation after an anomaly to determine the root cause of a problem.

In some case, measurements may involve active injection of test traffic to complete a measurement. However, most operators do not have access to user equipment, and injection of test traffic may be associated with costs in running such tests (e.g., the implications of bandwidth tests in a mobile network are obvious). Some active measurements (e.g., response under load or particular workloads) perturb other traffic, and could require dedicated access to the network segment. An alternative approach is to use in-network techniques that observe transport packet headers in operational networks to make the measurements.

In other cases, measurement involves dissecting network traffic flows. The observed transport layer information can help identify whether the link/network tuning is effective and alert to potential problems that can be hard to derive from link or device measurements alone. The design trade-offs for radio networks are often very different to those of wired networks. A radio-based network (e.g., cellular mobile, enterprise WiFi, satellite access/back-haul, point-to-point radio) has the complexity of a subsystem that performs radio resource management, with direct impact on the available capacity, and potentially loss/reordering of packets. The impact of the pattern of loss and congestion, differs for different traffic types, correlation with propagation and interference can all have significant impact on the cost and performance of a provided service. The need for this type of information is expected to increase as operators bring together heterogeneous types of network equipment and seek to deploy opportunistic methods to access radio spectrum.

3.4. Observing Headers to Implement Network Policy

Information from the transport protocol can be used by a multi-field classifier as a part of policy framework. Policies are commonly used for management of the QoS or Quality of Experience (QoE) in resource-constrained networks and by firewalls that use the information to implement access rules (see also section 2.2.2 of [RFC8404]). Traffic that cannot be classified, will typically receive a default treatment.

4. Encryption and Authentication of Transport Headers

End-to-end encryption can be applied at various protocol layers. It can be applied above the transport to encrypt the transport payload. Encryption methods can hide information from an eavesdropper in the network. Encryption can also help protect the privacy of a user, by

hiding data relating to user/device identity or location. Neither an integrity check nor encryption methods prevent traffic analysis, and usage needs to reflect that profiling of users, identification of location and fingerprinting of behaviour can take place even on encrypted traffic flows.

There are several motivations:

- o One motive to use encryption is a response to perceptions that the network has become ossified by over-reliance on middleboxes that prevent new protocols and mechanisms from being deployed. This has led to a perception that there is too much "manipulation" of protocol headers within the network, and that designing to deploy in such networks is preventing transport evolution. In the light of this, a method that authenticates transport headers may help improve the pace of transport development, by eliminating the need to always consider deployed middleboxes [I-D.trammell-plus-abstract-mech], or potentially to only explicitly enable middlebox use for particular paths with particular middleboxes that are deliberately deployed to realise a useful function for the network and/or users[RFC3135].
- o Another motivation stems from increased concerns about privacy and surveillance. Some Internet users have valued the ability to protect identity, user location, and defend against traffic analysis, and have used methods such as IPsec Encapsulated Security Payload (ESP), Virtual Private Networks (VPNs) and other encrypted tunnel technologies. Revelations about the use of pervasive surveillance [RFC7624] have, to some extent, eroded trust in the service offered by network operators, and following the Snowden revelation in the USA in 2013 has led to an increased desire for people to employ encryption to avoid unwanted "eavesdropping" on their communications. Concerns have also been voiced about the addition of information to packets by third parties to provide analytics, customization, advertising, cross-site tracking of users, to bill the customer, or to selectively allow or block content. Whatever the reasons, there are now activities in the IETF to design new protocols that may include some form of transport header encryption (e.g., QUIC [I-D.ietf-quic-transport]).

Authentication methods (that provide integrity checks of protocols fields) have also been specified at the network layer, and this also protects transport header fields. The network layer itself carries protocol header fields that are increasingly used to help forwarding decisions reflect the need of transport protocols, such as the IPv6 Flow Label [RFC6437], the DSCP and ECN.

The use of transport layer authentication and encryption exposes a tussle between middlebox vendors, operators, applications developers and users.

- o On the one hand, future Internet protocols that enable large-scale encryption assist in the restoration of the end-to-end nature of the Internet by returning complex processing to the endpoints, since middleboxes cannot modify what they cannot see.
- o On the other hand, encryption of transport layer header information has implications for people who are responsible for operating networks and researchers and analysts seeking to understand the dynamics of protocols and traffic patterns.

Whatever the motives, a decision to use pervasive of transport header encryption will have implications on the way in which design and evaluation is performed, and which can in turn impact the direction of evolution of the TCP/IP stack. While the IETF can specify protocols, the success in actual deployment is often determined by many factors [RFC5218] that are not always clear at the time when protocols are being defined.

The next subsections briefly review some security design options for transport protocols. A Survey of Transport Security Protocols [I-D.ietf-taps-transport-security] provides more details concerning commonly used encryption methods at the transport layer.

4.1. Authenticating the Transport Protocol Header

Transport layer header information can be authenticated. An integrity check that protects the immutable transport header fields, but can still expose the transport protocol header information in the clear, allowing in-network devices to observe these fields. An integrity check can not prevent in-network modification, but can avoid a receiving accepting changes and avoid impact on the transport protocol operation.

An example transport authentication mechanism is TCP-Authentication (TCP-AO) [RFC5925]. This TCP option authenticates the IP pseudo header, TCP header, and TCP data. TCP-AO protects the transport layer, preventing attacks from disabling the TCP connection itself and provides replay protection. TCP-AO may interact with middleboxes, depending on their behaviour [RFC3234].

The IPsec Authentication Header (AH) [RFC4302] was designed to work at the network layer and authenticate the IP payload. This approach authenticates all transport headers, and verifies their integrity at the receiver, preventing in-network modification.

4.2. Encrypting the Transport Payload

The transport layer payload can be encrypted to protect the content of transport segments. This leaves transport protocol header information in the clear. The integrity of immutable transport header fields could be protected by combining this with an integrity check (Section 4.1).

Examples of encrypting the payload include Transport Layer Security (TLS) over TCP [RFC5246] [RFC7525], Datagram TLS (DTLS) over UDP [RFC6347] [RFC7525], and TCPcrypt [I-D.ietf-tcpinc-tcpcrypt], which permits opportunistic encryption of the TCP transport payload.

4.3. Encrypting the Transport Header

The network layer payload could be encrypted (including the entire transport header and the payload). This method provides confidentiality of the entire transport packet. It therefore does not expose any transport information to devices in the network, which also prevents modification along a network path.

One example of encryption at the network layer is use of IPsec Encapsulating Security Payload (ESP) [RFC4303] in tunnel mode. This encrypts and authenticates all transport headers, preventing visibility of the transport headers by in-network devices. Some Virtual Private Network (VPN) methods also encrypt these headers.

4.4. Authenticating Transport Information and Selectively Encrypting the Transport Header

A transport protocol design can encrypt selected header fields, while also choosing to authenticate fields in the transport header. This allows specific transport header fields to be made observable by network devices. End-to-end integrity checks can prevent an endpoint from undetected modification of the immutable transport headers.

Mutable fields in the transport header provide opportunities for middleboxes to modify the transport behaviour (e.g., the extended headers described in [I-D.trammell-plus-abstract-mech]). This considers only immutable fields in the transport headers, that is, fields that may be authenticated End-to-End across a path.

An example of a method that encrypts some, but not all, transport information is GRE-in-UDP [RFC8086] when used with GRE encryption.

4.5. Optional Encryption of Header Information

There are implications to the use of optional header encryption in the design of a transport protocol, where support of optional mechanisms can increase the complexity of the protocol and its implementation and in the management decisions that are required to use variable format fields. Instead, fields of a specific type ought to always be sent with the same level of confidentiality or integrity protection.

5. Addition of Transport Information to Network-Layer Protocol Headers

Transport protocol information can be made visible in a network-layer header. This has the advantage that this information can then be observed by in-network devices. This has the advantage that a single header can support all transport protocols, but there may also be less desirable implications of separating the operation of the transport protocol from the measurement framework.

Some measurements may be made by adding additional protocol headers carrying operations, administration and management (OAM) information to packets at the ingress to a maintenance domain (e.g., an Ethernet protocol header with timestamps and sequence number information using a method such as 802.1lag or in-situ OAM [I-D.ietf-ippm-ioam-data]) and removing the additional header at the egress of the maintenance domain. This approach enables some types of measurements, but does not cover the entire range of measurements described in this document. In some cases, it can be difficult to position measurement tools at the required segments/nodes and there can be challenges in correlating the downstream/upstream information when in-band OAM data is inserted by an on-path device.

Another example of a network-layer approach is the IPv6 Performance and Diagnostic Metrics (PDM) Destination Option [RFC8250]. This allows a sender to optionally include a destination option that carries header fields that can be used to observe timestamps and packet sequence numbers. This information could be authenticated by receiving transport endpoints when the information is added at the sender and visible at the receiving endpoint, although methods to do this have not currently been proposed. This method needs to be explicitly enabled at the sender.

It can be undesirable to rely on methods requiring the presence of network options or extension headers. IPv4 network options are often not supported (or are carried on a slower processing path) and some IPv6 networks are also known to drop packets that set an IPv6 header extension (e.g., [RFC7872]). Another disadvantage is that protocols that separately expose header information do not necessarily have an

advantage to expose the information that is utilised by the protocol itself, and could manipulate this header information to gain an advantage from the network.

6. Implications of Protecting the Transport Headers

The choice of which fields to expose and which to encrypt is a design choice for the transport protocol. Any selective encryption method requires trading two conflicting goals for a transport protocol designer to decide which header fields to encrypt. Security work typically employs a design technique that seeks to expose only what is needed. However, there can be performance and operational benefits in exposing selected information to network tools.

This section explores key implications of working with encrypted transport protocols.

6.1. Independent Measurement

Independent observation by multiple actors is important for scientific analysis. Encrypting transport header encryption changes the ability for other actors to collect and independently analyse data. Internet transport protocols employ a set of mechanisms. Some of these need to work in cooperation with the network layer - loss detection and recovery, congestion detection and congestion control, some of these need to work only End-to-End (e.g., parameter negotiation, flow-control).

When encryption conceals information in the transport header, it could be possible for an applications to provide summary data on performance and usage of the network. This data could be made available to other actors. However, this data needs to contain sufficient detail to understand (and possibly reconstruct the network traffic pattern for further testing) and to be correlated with the configuration of the network paths being measured.

Sharing information between actors needs also to consider the privacy of the user and the incentives for providing accurate and detailed information. Protocols that expose the state information used by the transport protocol in their header information (e.g., timestamps used to calculate the RTT, packet numbers used to assess congestion and requests for retransmission) provide an incentive for the sending endpoint to provide correct information, increasing confidence that the observer understands the transport interaction with the network. This becomes important when considering changes to transport protocols, changes in network infrastructure, or the emergence of new traffic patterns.

6.2. Characterising "Unknown" Network Traffic

The patterns and types of traffic that share Internet capacity changes with time as networked applications, usage patterns and protocols continue to evolve.

If "unknown" or "uncharacterised" traffic patterns form a small part of the traffic aggregate passing through a network device or segment of the network the path, the dynamics of the uncharacterised traffic may not have a significant collateral impact on the performance of other traffic that shares this network segment. Once the proportion of this traffic increases, the need to monitor the traffic and determine if appropriate safety measures need to be put in place.

Tracking the impact of new mechanisms and protocols requires traffic volume to be measured and new transport behaviours to be identified. This is especially true of protocols operating over a UDP substrate. The level and style of encryption needs to be considered in determining how this activity is performed. On a shorter timescale, information may also need to be collected to manage denial of service attacks against the infrastructure.

6.3. Accountability and Internet Transport Protocols

Information provided by tools observing transport headers can be used to classify traffic, and to limit the network capacity used by certain flows. Operators can potentially use this information to prioritise or de-prioritise certain flows or classes of flow, with potential implications for network neutrality, or to rate limit malicious or otherwise undesirable flows (e.g., for Distributed Denial of Service, DDOS, protection, or to ensure compliance with a traffic profile Section 3.2.4). Equally, operators could use analysis of transport headers and transport flow state to demonstrate that they are not providing differential treatment to certain flows. Obfuscating or hiding this information using encryption is expected to lead operators and maintainers of middleboxes (firewalls, etc.) to seek other methods to classify, and potentially other mechanisms to condition, network traffic.

A lack of data reduces the level of precision with which flows can be classified and conditioning mechanisms are applied (e.g., rate limiting, circuit breaker techniques [RFC8084], or blocking of uncharacterised traffic), and this needs to be considered when evaluating the impact of designs for transport encryption [RFC5218].

6.4. Impact on Research, Development and Deployment

The majority of present Internet applications use two well-known transport protocols: e.g., TCP and UDP. Although TCP represents the majority of current traffic, some important real-time applications use UDP, and much of this traffic utilises RTP format headers in the payload of the UDP datagram. Since these protocol headers have been fixed for decades, a range of tools and analysis methods have become common and well-understood. Over this period, the transport protocol headers have mostly changed slowly, and so also the need to develop tools track new versions of the protocol.

Looking ahead, there will be a need to update these protocols and to develop and deploy new transport mechanisms and protocols. There are both opportunities and also challenges to the design, evaluation and deployment of new transport protocol mechanisms.

Integrity checks can protect an endpoint from undetected modification of protocol fields by network devices, whereas encryption and obfuscation can further prevent these headers being utilised by network devices. Hiding headers can therefore provide the opportunity for greater freedom to update the protocols and can ease experimentation with new techniques and their final deployment in endpoints.

Hiding headers can limit the ability to measure and characterise traffic. Measurement data is increasingly being used to inform design decisions in networking research, during development of new mechanisms and protocols and in standardisation. Measurement has a critical role in the design of transport protocol mechanisms and their acceptance by the wider community (e.g., as a method to judge the safety for Internet deployment). Observation of pathologies are also important in understanding the interactions between cooperating protocols and network mechanism, the implications of sharing capacity with other traffic and the impact of different patterns of usage.

Evolution and the ability to understand (measure) the impact need to proceed hand-in-hand. Attention needs to be paid to the expected scale of deployment of new protocols and protocol mechanisms. Whatever the mechanism, experience has shown that it is often difficult to correctly implement combination of mechanisms [RFC8085]. These mechanisms therefore typically evolve as a protocol matures, or in response to changes in network conditions, changes in network traffic or changes to application usage.

New transport protocol formats are expected to facilitate an increased pace of transport evolution, and with it the possibility to experiment with and deploy a wide range of protocol mechanisms.

There has been recent interest in a wide range of new transport methods, e.g., Larger Initial Window, Proportional Rate Reduction (PRR), congestion control methods based on measuring bottleneck bandwidth and round-trip propagation time, the introduction of AQM techniques and new forms of ECN response (e.g., Data Centre TCP, DCTP, and methods proposed for L4S). The growth and diversity of applications and protocols using the Internet also continues to expand. For each new method or application it is desirable to build a body of data reflecting its behaviour under a wide range of deployment scenarios, traffic load, and interactions with other deployed/candidate methods.

Open standards motivate a desire for this evaluation to include independent observation and evaluation of performance data, which in turn suggests control over where and when measurement samples are collected. This requires consideration of the appropriate balance between encrypting all and no transport information.

7. Conclusions

The majority of present Internet applications use two well-known transport protocols: e.g., TCP and UDP. Although TCP represents the majority of current traffic, some important real-time applications have used UDP, and much of this traffic utilises RTP format headers in the payload of the UDP datagram. Since these protocol headers have been fixed for decades, a range of tools and analysis methods have become common and well-understood. Over this period, the transport protocol headers have mostly changed slowly, and so also the need to develop tools track new versions of the protocol.

Confidentiality and strong integrity checks have properties that are being incorporated into new protocols and which have important benefits. The pace of development of transports using the WebRTC data channel and the rapid deployment of QUIC prototype transports can both be attributed to using a combination of UDP transport and confidentiality of the UDP payload.

The traffic that can be observed by on-path network devices is a function of transport protocol design/options, network use, applications and user characteristics. In general, when only a small proportion of the traffic has a specific (different) characteristic. Such traffic seldom leads to an operational issue although the ability to measure and monitor it is less. The desire to understand the traffic and protocol interactions typically grows as the proportion of traffic increases in volume. The challenges increase when multiple instances of an evolving protocol contribute to the traffic that share network capacity.

An increased pace of evolution therefore needs to be accompanied by methods that can be successfully deployed and used across operational networks. This leads to a need for network operators (at various level (ISPs, enterprises, firewall maintainer, etc) to identify appropriate operational support functions and procedures.

Protocols that change their transport header format (wire format) or their behaviour (e.g., algorithms that are needed to classify and characterise the protocol), will require new tooling needs to be developed to catch-up with the changes. If the currently deployed tools and methods are no longer relevant and performance may not be correctly measured. This can increase the response-time after faults, and can impact the ability to manage the network resulting in traffic causing traffic to be treated inappropriately (e.g., rate limiting because of being incorrectly classified/monitored). There are benefits in exposing consistent information to the network that avoids traffic being mis-classified and then receiving a default treatment by the network.

As a part of its design a new protocol specification therefore needs to weigh the benefits of ossifying common headers, versus the potential demerits of exposing specific information that could be observed along the network path to provide tools to manage new variants of protocols. Several scenarios to illustrate different ways this could evolve are provided below:

- o One scenario is when transport protocols provide consistent information to the network by intentionally exposing a part of the transport header. The design fixes the format of this information between versions of the protocol. This ossification of the transport header allows an operator to establish tooling and procedures that enable it to provide consistent traffic management as the protocol evolves. In contrast to TCP (where all protocol information is exposed), evolution of the transport is facilitated by providing cryptographic integrity checks of the transport header fields (preventing undetected middlebox changes) and encryption of other protocol information (preventing observation within the network, or incentivising the use of the exposed information, rather than inferring information from other characteristics of the flow traffic). The exposed transport information can be used by operators to provide troubleshooting, measurement and any necessary functions appropriate to the class of traffic (priority, retransmission, reordering, circuit breakers, etc).
- o An alternative scenario adopts different design goals, with a different outcome. A protocol that encrypts all header information forces network operators to act independently from

apps/transport developments to provide the transport information they need. A range of approaches may proliferate, as in current networks, operators can add a shim header to each packet as a flow as it crosses the network; other operators/managers could develop heuristics and pattern recognition to derive information that classifies flows and estimates quality metrics for the service being used; some could decide to rate-limit or block traffic until new tooling is in place. In many cases, the derived information can be used by operators to provide necessary functions appropriate to the class of traffic (priority, retransmission, reordering, circuit breakers, etc). Troubleshooting, and measurement becomes more difficult, and more diverse. This could require additional information beyond that visible in the packet header and when this information is used to inform decisions by on-path devices it can lead to dependency on other characteristics of the flow. In some cases, operators might need access to keying information to interpret encrypted data that they observe. Some use cases could demand use of transports that do not use encryption.

The outcome could have significant implications on the way the Internet architecture develops. It exposes a risk that significant actors (e.g., developers and transport designers) achieve more control of the way in which the Internet architecture develops. In particular, there is a possibility that designs could evolve to significantly benefit of customers for a specific vendor, and that communities with very different network, applications or platforms could then suffer at the expense of benefits to their vendors own customer base. In such a scenario, there could be no incentive to support other applications/products or to work in other networks leading to reduced access for new approaches.

8. Security Considerations

This document is about design and deployment considerations for transport protocols. Issues relating to security are discussed in the various sections of the document.

Authentication, confidentiality protection, and integrity protection are identified as Transport Features by [RFC8095]. As currently deployed in the Internet, these features are generally provided by a protocol or layer on top of the transport protocol [I-D.ietf-taps-transport-security].

Confidentiality and strong integrity checks have properties that can also be incorporated into the design of a transport protocol. Integrity checks can protect an endpoint from undetected modification of protocol fields by network devices, whereas encryption and

obfuscation can further prevent these headers being utilised by network devices. Hiding headers can therefore provide the opportunity for greater freedom to update the protocols and can ease experimentation with new techniques and their final deployment in endpoints. A protocol specification needs to weigh the benefits of ossifying common headers, versus the potential demerits of exposing specific information that could be observed along the network path to provide tools to manage new variants of protocols.

A protocol design that uses header encryption can provide confidentiality of some or all of the protocol header information. This prevents an on-path device from knowledge of the header field. It therefore prevents mechanisms being built that directly rely on the information or seeks to imply semantics of an exposed header field. Hiding headers can limit the ability to measure and characterise traffic.

Exposed transport headers are sometimes utilised as a part of the information to detect anomalies in network traffic. This can be used as the first line of defence to identify potential threats from DOS or malware and redirect suspect traffic to dedicated nodes responsible for DOS analysis, malware detection, or to perform packet scrubbing "Scrubbing" (the normalization of packets so that there are no ambiguities in interpretation by the ultimate destination of the packet). These techniques are currently used by some operators to also defend from distributed DOS attacks.

Exposed transport headers are sometimes also utilised as a part of the information used by the receiver of a transport protocol to protect the transport layer from data injection by an attacker. In evaluating this use of exposed header information, it is important to consider whether it introduces a significant DOS threat. For example, an attacker could construct a DOS attack by sending packets with a sequence number that falls within the currently accepted range of sequence numbers at the receiving endpoint, this would then introduce additional work at the receiving endpoint, even though the data in the attacking packet may not finally be delivered by the transport layer. This is sometimes known as a "shadowing attack". An attack can, for example, disrupt receiver processing, trigger loss and retransmission, or make a receiving endpoint perform unproductive decryption of packets that cannot be successfully decrypted (forcing a receiver to commit decryption resources, or to update and then restore protocol state).

One mitigation to off-path attack is to deny knowledge of what header information is accepted by a receiver or obfuscate the accepted header information, e.g., setting a non-predictable initial value for a sequence number during a protocol handshake, as in [RFC3550] and

[RFC6056], or a port value that can not be predicted (see section 5.1 of [RFC8085]). A receiver could also require additional information to be used as a part of check before accepting packets at the transport layer (e.g., utilising a part of the sequence number space that is encrypted; or by verifying an encrypted token not visible to an attacker). This would also mitigate on-path attacks. An additional processing cost can be incurred when decryption needs to be attempted before a receiver is able to discard injected packets.

Open standards motivate a desire for this evaluation to include independent observation and evaluation of performance data, which in turn suggests control over where and when measurement samples are collected. This requires consideration of the appropriate balance between encrypting all and no transport information. Open data, and accessibility to tools that can help understand trends in application deployment, network traffic and usage patterns can all contribute to understanding security challenges.

9. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

10. Acknowledgements

The authors would like to thank Mohamed Boucadair, Spencer Dawkins, Jana Iyengar, Mirja Kuehlewind, Kathleen Moriarty, Al Morton, Chris Seal, Joe Touch, Brian Trammell, and other members of the TSVWG for their comments and feedback.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421. The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

This work has received funding from the UK Engineering and Physical Sciences Research Council under grant EP/R04144X/1.

11. Informative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-03 (work in progress), June 2018.

- [I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-14 (work in progress), August 2018.
- [I-D.ietf-taps-transport-security]
Pauly, T., Perkins, C., Rose, K., and C. Wood, "A Survey of Transport Security Protocols", draft-ietf-taps-transport-security-02 (work in progress), June 2018.
- [I-D.ietf-tcpinc-tcpcrypt]
Bittau, A., Giffin, D., Handley, M., Mazieres, D., Slack, Q., and E. Smith, "Cryptographic protection of TCP Streams (tcpcrypt)", draft-ietf-tcpinc-tcpcrypt-12 (work in progress), June 2018.
- [I-D.ietf-tsvwg-l4s-arch]
Briscoe, B., Schepper, K., and M. Bagnulo, "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture", draft-ietf-tsvwg-l4s-arch-02 (work in progress), March 2018.
- [I-D.thomson-quic-grease]
Thomson, M., "More Apparent Randomization for QUIC", draft-thomson-quic-grease-00 (work in progress), December 2017.
- [I-D.trammell-plus-abstract-mech]
Trammell, B., "Abstract Mechanisms for a Cooperative Path Layer under Endpoint Control", draft-trammell-plus-abstract-mech-00 (work in progress), September 2016.
- [Latency] Briscoe, B., "Reducing Internet Latency: A Survey of Techniques and Their Merits", November 2014.
- [Measure] Fairhurst, G., Kuehlewind, M., and D. Lopez, "Measurement-based Protocol Design", June 2017.
- [RFC1273] Schwartz, M., "Measurement Study of Changes in Service-Level Reachability in the Global TCP/IP Internet: Goals, Experimental Design, Implementation, and Policy Considerations", RFC 1273, DOI 10.17487/RFC1273, November 1991, <<https://www.rfc-editor.org/info/rfc1273>>.

- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <<https://www.rfc-editor.org/info/rfc2914>>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<https://www.rfc-editor.org/info/rfc3135>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, DOI 10.17487/RFC3234, February 2002, <<https://www.rfc-editor.org/info/rfc3234>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<https://www.rfc-editor.org/info/rfc3550>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <<https://www.rfc-editor.org/info/rfc4585>>.

- [RFC4737] Morton, A., Ciavattone, L., Ramachandran, G., Shalunov, S., and J. Perser, "Packet Reordering Metrics", RFC 4737, DOI 10.17487/RFC4737, November 2006, <<https://www.rfc-editor.org/info/rfc4737>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC5236] Jayasumana, A., Piratla, N., Banka, T., Bare, A., and R. Whitner, "Improved Packet Reordering Metrics", RFC 5236, DOI 10.17487/RFC5236, June 2008, <<https://www.rfc-editor.org/info/rfc5236>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <<https://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, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <<https://www.rfc-editor.org/info/rfc6347>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.

- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <<https://www.rfc-editor.org/info/rfc7567>>.
- [RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <<https://www.rfc-editor.org/info/rfc7624>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC7928] Kuhn, N., Ed., Natarajan, P., Ed., Khademi, N., Ed., and D. Ros, "Characterization Guidelines for Active Queue Management (AQM)", RFC 7928, DOI 10.17487/RFC7928, July 2016, <<https://www.rfc-editor.org/info/rfc7928>>.
- [RFC8033] Pan, R., Natarajan, P., Baker, F., and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", RFC 8033, DOI 10.17487/RFC8033, February 2017, <<https://www.rfc-editor.org/info/rfc8033>>.
- [RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <<https://www.rfc-editor.org/info/rfc8084>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.

- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.
- [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <<https://www.rfc-editor.org/info/rfc8087>>.
- [RFC8095] Fairhurst, G., Ed., Trammell, B., Ed., and M. Kuehlewind, Ed., "Services Provided by IETF Transport Protocols and Congestion Control Mechanisms", RFC 8095, DOI 10.17487/RFC8095, March 2017, <<https://www.rfc-editor.org/info/rfc8095>>.
- [RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <<https://www.rfc-editor.org/info/rfc8250>>.
- [RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <<https://www.rfc-editor.org/info/rfc8257>>.
- [RFC8289] Nichols, K., Jacobson, V., McGregor, A., Ed., and J. Iyengar, Ed., "Controlled Delay Active Queue Management", RFC 8289, DOI 10.17487/RFC8289, January 2018, <<https://www.rfc-editor.org/info/rfc8289>>.
- [RFC8290] Hoeiland-Joergensen, T., McKenney, P., Taht, D., Gettys, J., and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", RFC 8290, DOI 10.17487/RFC8290, January 2018, <<https://www.rfc-editor.org/info/rfc8290>>.
- [RFC8404] Moriarty, K., Ed. and A. Morton, Ed., "Effects of Pervasive Encryption on Operators", RFC 8404, DOI 10.17487/RFC8404, July 2018, <<https://www.rfc-editor.org/info/rfc8404>>.

Appendix A. Revision information

-00 This is an individual draft for the IETF community.

-01 This draft was a result of walking away from the text for a few days and then reorganising the content.

-02 This draft fixes textual errors.

-03 This draft follows feedback from people reading this draft.

-04 This adds an additional contributor and includes significant reworking to ready this for review by the wider IETF community Colin Perkins joined the author list.

Comments from the community are welcome on the text and recommendations.

-05 Corrections received and helpful inputs from Mohamed Boucadair.

-06 Updated following comments from Stephen Farrell, and feedback via email. Added a draft conclusion section to sketch some strawman scenarios that could emerge.

-07 Updated following comments from Al Morton, Chris Seal, and other feedback via email.

-08 Updated to address comments sent to the TSVWG mailing list by Kathleen Moriarty (on 08/05/2018 and 17/05/2018), Joe Touch on 11/05/2018, and Spencer Dawkins.

-09 Updated security considerations.

-10 Updated references, split the Introduction, and added a paragraph giving some examples of why ossification has been an issue.

Authors' Addresses

Godred Fairhurst
University of Aberdeen
Department of Engineering
Fraser Noble Building
Aberdeen AB24 3UE
Scotland

EMail: gorry@erg.abdn.ac.uk
URI: <http://www.erg.abdn.ac.uk/>

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
Scotland

EMail: csp@csperskins.org
URI: <https://csperskins.org/>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 12, 2018

B. Trammell
M. Kuehlewind
ETH Zurich
April 10, 2018

The Wire Image of a Network Protocol
draft-trammell-wire-image-04

Abstract

This document defines the wire image, an abstraction of the information available to an on-path non-participant in a networking protocol. This abstraction is intended to shed light on the implications on increased encryption has for network functions that use the wire image.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 12, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

A protocol specification defines a set of behaviors for each participant in the protocol: which lower-layer protocols are used for which services, how messages are formatted and protected, which participant sends which message when, how each participant should respond to each message, and so on.

Implicit in a protocol specification is the information the protocol radiates toward nonparticipant observers of the messages sent among participants, often including participants in lower layer protocols. Any information that has a clear definition in the protocol's message format(s), or is implied by that definition, and is not cryptographically confidentiality-protected can be unambiguously interpreted by those observers.

This information comprises the protocol's wire image, which we define and discuss in this document. It is the wire image, not the protocol's specification, that determines how third parties on the network paths among protocol participants will interact with that protocol.

Several documents currently under discussion in IETF working groups and the IETF in general, for example [QUIC-MANAGEABILITY], [EFFECT-ENCRYPT], and [TRANSPORT-ENCRYPT], discuss in part impacts on the third-party use of wire images caused by a migration from protocols whose wire images are largely not confidentiality protected (e.g. HTTP over TCP) to protocols whose wire images are confidentiality protected (e.g. H2 over QUIC).

This document presents the wire image abstraction with the hope that it can shed some light on these discussions.

2. Definition

More formally, the wire image of a protocol consists of the sequence of messages sent by each participant in the protocol, each expressed as a sequence of bits with an associated arbitrary-precision time at which it was sent.

3. Discussion

This definition is so vague as to be difficult to apply to protocol analysis, but it does illustrate some important properties of the wire image.

Key is that the wire image is not limited to merely "the unencrypted bits in the header". In particular, interpacket timing, packet size,

and message sequence information can be used to infer other parameters of the behavior of the protocol, or to fingerprint protocols and/or specific implementations of the protocol; see Section 3.1.

An important implication of this property is that a protocol which uses confidentiality protection for the headers it needs to operate can be deliberately designed to have a specified wire image that is separate from that machinery; see Section 3.3. Note that this is a capability unique to encrypted protocols. Parts of a wire image may also be made visible to devices on path, but immutable through end-to-end integrity protection; see Section 3.2.

Portions of the wire image of a protocol that are neither confidentiality-protected nor integrity-protected are writable by devices on the path(s) between the endpoints using the protocol. A protocol with a wire image that is largely writable operating over a path with devices that understand the semantics of the protocol's wire image can modify it, in order to induce behaviors at the protocol's participants. This is the case with TCP in the current Internet.

Note also that the wire image is multidimensional. This implies that the name "image" is not merely metaphorical, and that general image recognition techniques may be applicable to extracting patterns and information from it.

From the point of view of a passive observer, the wire image of a single protocol is rarely seen in isolation. The dynamics of the application and network stacks on each endpoint use multiple protocols for any higher level task. Most protocols involving user content, for example, are often seen on the wire together with DNS traffic; the information from these two wire images can be correlated to infer information about the dynamics of the overlying application.

3.1. Obscuring timing and sizing information

Cryptography can protect the confidentiality of a protocol's headers, to the extent that forwarding devices do not need the confidentiality-protected information for basic forwarding operations. However, it cannot be applied to protecting non-header information in the wire image. Of particular interest is the sequence of packet sizes and the sequence of packet times. These are characteristic of the operation of the protocol. While packets cannot be made smaller than their information content, nor sent faster than processing time requirements at the sender allow, a sender may use padding to increase the size of packets, and add delay to transmission scheduling in order to increase interpacket delay.

However, it does this at the expense of bandwidth efficiency and latency, so this technique is limited to the application's tolerance for latency and bandwidth inefficiency.

3.2. Integrity Protection of the Wire Image

Adding end-to-end integrity protection to portions of the wire image makes it impossible for on-path devices to modify them without detection by the endpoints, which can then take action in response to those modifications, making these portions of the wire image effectively immutable. However, they can still be observed by devices on path. This allows the creation of signals intended by the endpoints solely for the consumption of these on-path devices.

Integrity protection can only practically be applied to the sequence of bits in each packet, which implies that a protocol's visible wire image cannot be made completely immutable in a packet-switched network. Interarrival timings, for instance, cannot be easily protected, as the observable delay sequence is modified as packets move through the network and experience different delays on different links. Message sequences are also not practically protectable, as packets may be dropped or reordered at any point in the network, as a consequence of the network's operation. Intermediate systems with knowledge of the protocol semantics in the readable portion of the wire image can also purposely delay or drop packets in order to affect the protocol's operation.

3.3. Engineering the Wire Image

Understanding the nature of a protocol's wire image allows it to be engineered. The general principle at work here, observed through experience with deployability and non-deployability of protocols at the network and transport layers in the Internet, is that all observable parts of a protocol's wire image will eventually be used by devices on path; consequently, changes or future extensions that affect the observable part of the wire image become difficult or impossible to deploy.

A network function which serves a purpose useful to its deployer will use the information it needs from the wire image, and will tend to get that information from the wire image in the simplest way possible.

For example, consider the case of the ubiquitous TCP [RFC0793] transport protocol. As described in [PATH-SIGNALS], several key in-network functions have evolved to take advantage of implicit signals in TCP's wire image, which, as TCP provides neither integrity or

confidentiality protection for its headers, is inseparable from its internal operation. Some of these include:

- o Determining return routability and consent: For example, TCP's wire image contains both an implicit indication that the sender of a packet is at least on the path toward its source address (in the acknowledgement number during the handshake), as well as an implicit indication that a receiving device consents to continue communication. These are used by stateful network firewalls.
- o Measuring loss and latency: For example, examining the sequence of TCP's sequence and acknowledgement numbers, as well as the ECN [RFC3168] control bits allows the inference of congestion, loss and retransmission along the path. The sequence and acknowledgement numbers together with the timestamp option [RFC7323] allow the measurement of application-experienced latency.

During the design of a protocol, the utility of features such as these should be considered, and the protocol's wire image should therefore be designed to explicitly expose information to those network functions deemed important by the designers in an obvious way. The wire image should expose as little other information as possible.

However, even when information is explicitly provided to the network, any information that is exposed by the wire image, even that information not intended to be consumed by an observer, must be designed carefully as it might ossify, making it immutable for future versions of the protocol. For example, information needed to support decryption by the receiving endpoint (cryptographic handshakes, sequence numbers, and so on) may be used by devices along the path for their own purposes.

3.3.1. Declaring Protocol Invariants

One approach to reduce the extent of the wire image that will be used by devices on the path is to define a set of invariants for a protocol during its development. Declaring a protocol's invariants represents a promise made by the protocol's developers that certain bits in the wire image, and behaviors observable in the wire image, will be preserved through the specification of all future versions of the protocol. QUIC's invariants [QUIC-INVARIANTS] are an initial attempt to apply this approach to QUIC.

While static aspects of the wire image - bits with simple semantics at fixed positions in protocol headers - can easily be made invariant, different aspects of the wire image may be more or less

appropriate to define as invariants. For a protocol with a version and/or extension negotiation mechanism, the bits in the header and behaviors tied to those bits which implement version negotiation should be made invariant. More fluid aspects of the wire image and behaviors which are not necessary for interoperability are not appropriate as invariants.

Parts of a protocol's wire image not declared invariant but intended to be visible to devices on path should be protected against "accidental invariance": the deployment of on-path devices over time that make simplifying assumptions about the behavior of those parts of the wire image, making new behaviors not meeting those assumptions difficult to deploy. Integrity protection of the wire image may itself help protect against accidental invariance, because read-only wire images invite less meddling than path-writable wire images. The techniques discussed in [USE-IT] may also be useful in further preventing accidental invariance and ossification.

Likewise, parts of a protocol's wire image not declared invariant and not intended to be visible to the path should be encrypted to protect their confidentiality. When confidentiality protection is either not possible or not practical, then, as above, the approaches discussed in [USE-IT] may be useful in ossification prevention.

3.3.2. Trustworthiness of Engineered Signals

Since they are separate from the signals that drive an encrypted protocol's mechanisms, the veracity of integrity-protected signals in an engineered wire image intended for consumption by the path may not be verifiable by on-path devices; see [PATH-SIGNALS]. Indeed, any two endpoints with a secret channel between them (in this case, the encrypted protocol itself) may collude to change the semantics and information content of these signals. This is an unavoidable consequence of the separation of the wire image from the protocol's operation afforded by confidentiality protection of the protocol's headers.

4. Acknowledgments

Thanks to Martin Thomson, Thomas Fossati, Ted Hardie, Mark Nottingham, and the membership of the IAB Stack Evolution Program, for text, feedback, and discussions that have improved this document.

This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat for Education, Research, and Innovation under contract no. 15.0268. This support does not imply endorsement.

5. Informative References

[EFFECT-ENCRYPT]

Moriarty, K. and A. Morton, "Effects of Pervasive Encryption on Operators", draft-mm-wg-effect-encrypt-25 (work in progress), March 2018.

[PATH-SIGNALS]

Hardie, T., "Path Signals", draft-hardie-path-signals-03 (work in progress), April 2018.

[QUIC-INVARIANTS]

Thomson, M., "Version-Independent Properties of QUIC", draft-ietf-quic-invariants-01 (work in progress), March 2018.

[QUIC-MANAGEABILITY]

Kuehlewind, M. and B. Trammell, "Manageability of the QUIC Transport Protocol", draft-ietf-quic-manageability-01 (work in progress), October 2017.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <<https://www.rfc-editor.org/info/rfc7323>>.

[TRANSPORT-ENCRYPT]

Fairhurst, G. and C. Perkins, "The Impact of Transport Header Confidentiality on Network Operation and Evolution of the Internet", draft-fairhurst-tsvwg-transport-encrypt-07 (work in progress), April 2018.

[USE-IT]

Thomson, M., "Long-term Viability of Protocol Extension Mechanisms", draft-thomson-use-it-or-lose-it-01 (work in progress), March 2018.

Authors' Addresses

Brian Trammell
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: ietf@trammell.ch

Mirja Kuehlewind
ETH Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Email: mirja.kuehlewind@tik.ee.ethz.ch