

Workgroup: Network Working Group
Internet-Draft:
draft-ietf-masque-ip-proxy-reqs-03
Published: 27 August 2021
Intended Status: Informational
Expires: 28 February 2022
Authors: A. Chernyakhovsky D. McCall D. Schinazi
 Google LLC Google LLC Google LLC
Requirements for a MASQUE Protocol to Proxy IP Traffic

Abstract

There is interest among MASQUE working group participants in designing a protocol that can proxy IP traffic over HTTP. This document describes the set of requirements for such a protocol.

Discussion of this work is encouraged to happen on the MASQUE IETF mailing list masque@ietf.org or on the GitHub repository which contains the draft: <https://github.com/ietf-wg-masque/draft-ietf-masque-ip-proxy-reqs>.

Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-masque/draft-ietf-masque-ip-proxy-reqs>.

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 28 February 2022.

Copyright Notice

Copyright (c) 2021 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 |
| 1.1. | Conventions |
| 1.2. | Definitions |
| 2. | Use Cases |
| 2.1. | Consumer VPN |
| 2.2. | Point to Point Connectivity |
| 2.3. | Point to Network Connectivity |
| 3. | Requirements |
| 3.1. | IP Session Establishment |
| 3.2. | Proxying of IP packets |
| 3.3. | Maximum Transmission Unit |
| 3.4. | IP Assignment |
| 3.5. | Identity |
| 3.6. | Transport Security |
| 3.7. | Flow Control |
| 3.8. | Indistinguishability |
| 3.9. | Support HTTP/2 and HTTP/3 |
| 3.10. | Multiplexing |
| 3.11. | Statefulness |
| 4. | Extensibility |
| 4.1. | Authentication |
| 4.2. | Reliable Transmission of IP Packets |
| 4.3. | Configuration of Congestion and Flow Control |
| 4.4. | Data Transport Compression |
| 5. | Non-requirements |
| 5.1. | Addressing Architecture |
| 5.2. | Translation |
| 5.3. | IP Packet Extraction |
| 5.4. | Trust |
| 6. | Security Considerations |
| 7. | IANA Considerations |
| | Acknowledgments |
| | References |
| | Normative References |
| | Informative References |
| | Authors' Addresses |

1. Introduction

There exist several IETF standards for proxying IP in a way that is authenticated and confidential, such as IKEv2/IPsec [[IKEV2](#)]. However, those are distinguishable from common Internet traffic and often blocked. Additionally, large server deployments have expressed interest in using a VPN solution that leverages existing security protocols such as QUIC [[QUIC](#)] or TLS [[TLS](#)] to avoid adding another protocol to their security posture.

This document describes the set of requirements for a protocol that can proxy IP traffic over HTTP. The requirements outlined below are similar to the considerations made in designing the CONNECT-UDP method [[CONNECT-UDP](#)], additionally including IP-specific requirements, such as a means of negotiating the routes that should be advertised on either end of the connection.

Discussion of this work is encouraged to happen on the MASQUE IETF mailing list masque@ietf.org or on the GitHub repository which contains the draft: <https://github.com/ietf-wg-masque/draft-ietf-masque-ip-proxy-reqs>.

1.1. Conventions

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.

1.2. Definitions

*Data Transport: The mechanism responsible for transmitting IP packets over HTTP. This can involve streams or datagrams.

*IP Session: An association between client and server whereby both agree to proxy IP traffic given certain configuration properties. This is similar to a Child Security Association in IKEv2 terminology. An IP Session uses Data Transports to transmit packets.

2. Use Cases

There are multiple reasons to deploy an IP proxying protocol. This section discusses some examples of use cases that MUST be supported by the protocol. Note that while the protocol needs to support these use cases, the protocol elements that allow them may be optional.

2.1. Consumer VPN

Consumer VPNs refer to network applications that allow a user to hide some properties of their traffic from some network observers. In particular, it can hide the identity of servers the client is connecting to from the client's network provider, and can hide the client's IP address (and derived geographical information) from the servers they are communicating with. Note that this hidden information is now available to the VPN service provider, so is only beneficial for clients who trust the VPN service provider more than other entities.

2.2. Point to Point Connectivity

Point-to-point connectivity creates a private, encrypted and authenticated network between two IP addresses. This is useful, for example, with container networking to provide a virtual (overlay) network with addressing separate from the physical transport. An example of this is Wireguard.

2.3. Point to Network Connectivity

Point-to-Network connectivity is the more traditional remote-access "VPN" use case, frequently used when a user needs to connect to a different network (such as an enterprise network) for access to resources that are not exposed to the public Internet.

3. Requirements

This section lists requirements for a protocol that can proxy IP over an HTTP connection.

3.1. IP Session Establishment

The protocol will allow the client to request establishment of an IP Session, along with configuration options and one or more associated Data Transports. The server will have the ability to accept or deny the client's request.

3.2. Proxying of IP packets

The protocol will establish Data Transports, which will be able to forward IP packets. The Data Transports **MUST** be able to take IP datagrams input on one side and egress them unmodified in their entirety on the other side, although extensions may enable IP packets to be modified in transit. The protocol will support both IPv6 [[IPV6](#)] and IPv4 [[IPV4](#)].

3.3. Maximum Transmission Unit

The protocol will allow tunnel endpoints to inform each other of the Maximum Transmission Unit (MTU) they are willing to forward. This will allow avoiding some IP fragmentation, especially as IPv6 does not allow IP fragmentation by nodes along the path. In cases where the tunnel endpoint is not the same as the communication endpoint, tunnel endpoints are expected to apply the guidance on UDP tunnels in [[TUNNELS](#)].

3.4. IP Assignment

The client will be able to request to be assigned an IP address range, optionally specifying a preferred range. In response to that request, the server will either assign a range of its choosing to the client, or decline the request. For symmetry, the server may request assignment of an IP address range from the client, and the client will either assign a range or decline the request. Endpoints will also have the ability to assign an IP address range to their peer, and to communicate that assignment to the peer, without having received a request.

3.5. Identity

When negotiating the creation of an IP Session, the protocol will allow both endpoints to exchange an identifier. As examples, the identity could be a user name, an email address, a token, or a fully-qualified domain name. Note that this requirement does not cover authenticating the identifier.

3.6. Transport Security

The protocol **MUST** be run over a protocol that provides mutual authentication, confidentiality and integrity. Using QUIC or TLS would meet this requirement.

3.7. Flow Control

The protocol will allow the ability to proxy IP packets without flow control, at least when HTTP/3 is in use. QUIC DATAGRAM frames are not flow controlled and would meet this requirement. The document defining the protocol will provide guidance on how best to use flow control to improve IP Session performance.

3.8. Indistinguishability

A passive network observer not participating in the encrypted connection should not be able to distinguish IP proxying from regular encrypted HTTP Web traffic by only observing non-encrypted parts of the traffic. Specifically, any data sent unencrypted (such

as headers, or parts of the handshake) should look like the same unencrypted data that would be present for Web traffic. Traffic analysis is out of scope for this requirement.

3.9. Support HTTP/2 and HTTP/3

The IP proxying protocol discussed in this document will run over HTTP. The protocol SHOULD strongly prefer to use HTTP/3 [[H3](#)] and SHOULD use the QUIC DATAGRAM frames [[DGRAM](#)] when available to improve performance. The protocol MUST also support HTTP/2 [[H2](#)] as a fallback when UDP is blocked on the network path. Proxying IP over HTTP/2 MAY result in lower performance than over HTTP/3.

3.10. Multiplexing

Since recent HTTP versions support concurrently running multiple requests over the same connection, the protocol SHOULD support multiple independent instances of IP proxying over a given HTTP connection.

3.11. Statefulness

The protocol should limit the amount of state a MASQUE client or server needs to operate. Keeping minimal state simplifies reconnection in the presence of failures and can facilitate extensibility.

4. Extensibility

The protocol will provide a mechanism by which clients and servers can add extension information to the exchange that establishes the IP Session. If the solution uses an HTTP request and response, this could be accomplished using HTTP headers.

Once the IP Session is established, the protocol will provide a mechanism that allows reliably exchanging extension messages in both directions at any point in the lifetime of the IP Session.

The subsections below list possible extensions that designers of the protocol will keep in mind to ensure it will be possible to design such extensions.

4.1. Authentication

Since the protocol will offer a way to convey identity, extensions will allow authenticating that identity, from both the client and server, during the establishment of the IP Session. For example, an extension could allow a client to offer an OAuth Access Token [[OAUTH](#)] when requesting an IP Session. As another example, another

extension could allow an endpoint to demonstrate knowledge of a cryptographic secret.

4.2. Reliable Transmission of IP Packets

While it is desirable to transmit IP packets unreliably in most cases, an extension could provide a mechanism to allow forwarding some packets reliably. For example, when using HTTP/3, this can be accomplished by allowing Data Transports to run over both DATAGRAM and STREAM frames.

4.3. Configuration of Congestion and Flow Control

An extension will allow exchanging congestion and flow control parameters to improve performance. For example, an extension could disable congestion control for non-retransmitted Data Transports if it knows that the proxied traffic is itself congestion-controlled.

4.4. Data Transport Compression

While the core protocol Data Transports will transmit IP packets in their unmodified entirety, an extension can allow compressing these packets.

5. Non-requirements

This section discusses topics that are explicitly out of scope for the IP Proxying protocol. These topics MAY be handled by implementers or future extensions.

5.1. Addressing Architecture

This document only describes the requirements for a protocol that allows IP proxying. It does not discuss how the IPs assigned are determined, managed, or translated. While these details are important for producing a functional system, they do not need to be handled by the protocol beyond the ability to convey those assignments.

Similarly, "ownership" of an IP range is out of scope. If an endpoint communicates to its peer that it can allocate addresses from a range, or route traffic to a range, the peer has no obligation to trust that information. Whether or not to trust this information is left to individual implementations and extensions: the protocol will be extensible enough to allow the development of extensions that assist in assessing this trust.

5.2. Translation

Some servers may wish to perform Network Address Translation (NAT) or any other modification to packets they forward. Doing so is out of scope for the proxying protocol. In particular, the ability to discover the presence of a NAT, negotiate NAT bindings, or check connectivity through a NAT is explicitly out of scope and left to future extensions.

Servers that do not perform NAT will commonly forward packets similarly to how a traditional IP router would, but the specific of that are considered out of scope. In particular, decrementing the Hop Limit (or TTL) field of the IP header is out of scope for MASQUE and expected to be performed by a router behind the MASQUE server, or collocated with it.

5.3. IP Packet Extraction

How packets are forwarded between the IP proxying connection and the physical network is out of scope. For example, this can be accomplished on some operating systems using a TUN interface. How this is done is deliberately not specified and will be left to individual implementations.

5.4. Trust

All the use-cases described in [Section 2](#) require some level of trust between endpoints. However, how this trust is established and what decisions endpoints make based on this trust is considered out of scope. For example, if an endpoint doesn't sufficiently trust its peer, it would be well advised to validate the IP addresses used by that peer - however that is considered out of scope for the document that will describe an IP proxying protocol.

6. Security Considerations

This document only discusses requirements on a protocol that allows IP proxying. That protocol will need to document its security considerations.

7. IANA Considerations

This document requests no actions from IANA.

Acknowledgments

The authors would like to thank participants of the MASQUE working group for their feedback.

References

Normative References

- [DGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-datagram-03, 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-datagram-03>>.
- [H2] Belshé, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/rfc/rfc7540>>.
- [H3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, 2 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>>.
- [IPV4] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/rfc/rfc791>>.
- [IPV6] 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/rfc/rfc8200>>.
- [QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-34, 14 January 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34>>.
- [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/rfc/rfc2119>>.
- [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/rfc/rfc8174>>.
- [TLS] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/rfc/rfc8446>>.

Informative References

[CONNECT-UDP]

Schinazi, D., "The CONNECT-UDP HTTP Method", Work in Progress, Internet-Draft, draft-ietf-masque-connect-udp-04, 12 July 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-udp-04>>.

[IKEV2]

Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<https://www.rfc-editor.org/rfc/rfc7296>>.

[OAUTH]

Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <<https://www.rfc-editor.org/rfc/rfc6749>>.

[TUNNELS]

Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <<https://datatracker.ietf.org/doc/html/draft-ietf-intarea-tunnels-10>>.

Authors' Addresses

Alex Chernyakhovsky
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043,
United States of America

Email: achernya@google.com

Dallas McCall
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043,
United States of America

Email: dallasmccall@google.com

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043,
United States of America

Email: dschinazi.ietf@gmail.com