Workgroup: Network Working Group
Internet-Draft:
draft-cms-masque-ip-proxy-reqs-01
Published: 7 August 2020
Intended Status: Informational
Expires: 8 February 2021
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/DavidSchinazi/masque-drafts.

Status of This Memo

Copyright Notice

Table of Contents

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/DavidSchinazi/masque-drafts.

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 method by which IP packets are transmitted. 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.

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.

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.

2.4. Network to Network Connectivity

Network-to-Network connectivity is also called a site-to-site VPN. Like the point-to-network use case, the goal is to connect to a network that is not exposed publicly. The site-to-site aspects make this transparent to the user; the entire networks are connected to each other and route packets transparently without a VPN client installed on the user's device. This style of connectivity can also be used to connect devices that cannot run VPN clients through to the network.

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, in their unmodified entirety. The protocol will support both IPv6 [IPV6] and IPv4 [IPV4].

3.3. Maximum Transmission Unit

The protocol will allow endpoints to inform each other of the Maximum Transmission Unit (MTU) they are willing to forward. This will allow avoiding IP fragmentation, especially as IPv6 does not allow IP fragmentation by nodes along the path.

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. Similarly, to support the network-to-network use case, the server will be able to request assignment of an IP address range from the client, and the client will either assign a range or decline the request.

3.5. Route Negotiation

At any point in an IP Session (not limited to its initial negotiation), the protocol will allow both client and server to inform its peer that it can route a set of IP prefixes. Both endpoints can also request a route to a given prefix, and the peer can choose to provide that route or not.

3.6. Identity

When negotiating the creation of an IP Session, the protocol will allow both endpoints to exchange an identifier. For example, both endpoints will be able to identify themselves by sending a fully-qualified domain name. Note that the Identity requirement does not cover authenticating the identifier; that requirement is covered by Section 3.8.

3.7. 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.

3.8. Authentication

Additionally to the authentication provided by the transport, the protocol will have the ability to authenticate both client and server during the establishment of the IP Session. In particular, it will be possible for the client to offer an OAuth Access Token [OAUTH] to the server when requesting IP proxying, potentially through an extension of the protocol. The protocol will also have the ability to support vendor-specific authentication mechanisms as extensions.

3.9. Reliable Transmission of IP Packets

While it is desirable to transmit IP packets unreliably in most cases, the protocol will 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.

3.10. 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.

3.11. Indistinguishability

A passive network observer not participating in the encrypted connection should not be able to distinguish an IP proxying session from regular encrypted HTTP Web 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.

3.12. 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 SHOULD 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.

3.13. 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.

3.14. Load balancing

Clients and servers should each be able to instantiate new Data Transports. This facilitates multi-threaded servers being able to handle a higher bandwidth of IP proxied packets.

The IP proxying mechanisms need to support load balancing of the traffic sent across the session, such as to another server. The document defining the new protocol should provide guidance for when additional connections and/or sessions should be opened, as opposed to reusing existing ones.

3.15. 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 session is established, the protocol will provide a mechanism that allows reliably exchanging vendor-specific messages in both directions at any point in the lifetime of the IP Session.

4. 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.

4.1. 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.

4.2. 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 traditionnal 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.

4.3. IP Packet Extraction

How packets are forwarded between the IP proxying connection and the physical network is out of scope. This is deliberately not specified and will be left to individual implementations.

5. Security Considerations

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

6. 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-00, 26 February 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-datagram-00.txt>.

[H2] Belshe, 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/info/rfc7540>.

[H3] Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-29, 9 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-http-29.txt>.

[IPV4] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/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/info/rfc8200>.

[QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Work in Progress, Internet-Draft, draft-ietf-quic-transport-29, 9 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-quic-transport-29.txt>.

[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>.

[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>.

[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/info/rfc8446>.

Informative References

[CONNECT-UDP] Schinazi, D., "The CONNECT-UDP HTTP Method", Work in Progress, Internet-Draft, draft-schinazi-masque-connect-udp-00, 16 April 2020, <http://www.ietf.org/internet-drafts/draft-schinazi-masque-connect-udp-00.txt>.

[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/info/rfc7296>.

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

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