Network Working Group
Internet-Draft
Updates: 7983 (if approved)
Intended status: Informational
Expires: June 6, 2018

M. Thomson
Mozilla

December 03, 2017

Principles for Multiplexing of UDP-based Protocols
draft-thomson-mux-principles-00

Abstract

The existence of protocols that rely on multiplexing of different protocols could be used to justify the imposition of constraints on the operation of other protocols. The existence of a multiplexed protocol does not inherently justify constraints on protocols that participate or might participate in the protocol.

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

The use of multiple protocols that share a 5-tuple is possible in unordered transport like UDP. Multiplexing in this fashion creates a meta-protocol that is comprised of the set of protocols that are multiplexed.

A specific example of such a meta-protocol is documented in RFC 7983 [RT-MUX]. This protocol comprises up to 5 different protocols: STUN [STUN], ZRTP [ZRTP], DTLS [DTLS], TURN channels [TURN], and RTP [RTP]. The meta-protocol that is composed from this set of protocols is commonly deployed for use in real-time communications. Of particular note is the STUN protocol [STUN], which is specifically designed to be multiplexed with another protocol.

The existence of a multiplexed meta-protocol can be used to justify constraints on the definition of new protocols, or the addition of changes to protocols that participate. These constraints should be considered in the context of the protocol in use. A protocol design for use outside of a multiplexed context should not be unduly constrained by the limitations of a chosen multiplexing scheme.

2. Multiplexing is Almost Always Possible

Any protocol that includes integrity checks can be multiplexed with any other protocol. A sufficiently strong checksum or cryptographic authentication tag allows arriving datagrams to be rejected by any protocol that the datagram was not intended for with high probability.

New protocols often require confidentiality and integrity and so use a solution like TLS [TLS]. Other protocols benefit greatly from being robust against errors in the relatively weak checksum [CHECKSUM] provided by internet protocols [CHECKSUM-FAIL]. Thus, the inclusion of strong integrity checks is beneficial for any internet protocol, which makes the protocol highly likely to be classified by a recipient as intended.

Note: This does not mean that protocols are distinguishable to intermediaries. Protocols that include use keyed integrity checks will not be identifiable to entities that do not have access to keys.

Relying on an integrity check also probabilistic, meaning that shorter integrity checks increase the chance that a datagram could be

incorrectly classified by a recipient. While the probability of collision is negligible for a protocol that uses an integrity check with 128 bits or more of redundancy, a shorter integrity check could be vulnerable to collisions and mis-classification.

As long as no more than one participating protocol lacks an integrity check of sufficient strength, protocols can be demultiplexed with high reliability. However, this form of demultiplexing can be grossly inefficient, as every datagram needs to be validated once for each potential protocol that is in use.

3. Multiplexing Considerations

Practical multiplexing schemes rely on simpler and more direct differences between protocols. For instance, RFC 7983 [RT-MUX] separates protocols based on the value of the first octet. While this scheme is cheap and effective, it has the drawback of using a limited space of potential values. Of the 256 possible values for the first octet, RFC 7983 consumes 132 values. Adding more protocols to the set that can be multiplexed would reduce the available space further.

The design in RFC 7983 was initially opportunistic in nature. The original set of protocols that were selected included only DTLS, SRTP, and STUN. Since then, compatibility with that scheme has constrained the design of other protocols so that they fit within this scheme. This is not an indefinitely sustainable posture.

The need to multiplex can create unwanted constraints on the designs of protocols, particularly as protocols evolve. For a protocol that is used exclusively or predominantly in a multiplexed meta-protocol, this does not present a significant burden, but protocols that have uses in other contexts are different.

For instance, DTLS 1.3 [DTLS13] defines a record layout that uses the first octet in a very different fashion to earlier versions; this design is constrained by the need to remain compatible with the scheme in RFC 7983 [RT-MUX].

Multiplexing should not be a primary consideration in design of new protocols that might be multiplexed. Similary, the design of extensions or revisions to protocols should not be constrained by the potential for multiplexing. Multiplexing considerations might be paid greater attention depending on how likely it is that the protocol will be used together with other protocols.

For example, a new version of RTP [RTP] might consider setting the version field to 3. While this space in the multiplexing scheme in

[RT-MUX] is currently unallocated, changing the RTP version greatly reduces the space available for other protocols. Because RTP is frequently used in a multiplexed context, changing the RTP header is best avoided.

3.1. Alternative Multiplexing Designs

Protocol multiplexing works most efficiently when the protocol in use for any given packet can be identified using only the contents of the packet. Stateful multiplexing - where multiplexing rules change based on protocol state - is possible, but is best avoided.

A protocol that intends to multiplex several protocols can avoid this sort of pressure by adding an explicit multiplexing protocol layer. The simplest multiplexing protocol is a shim of a small number of octets that is added to the start or end of every datagram. The value of these octets is then used to identify the protocol.

For a scheme like that in [RT-MUX], it is possible to use a shim for only some protocols, whether other protocols are multiplexed based on their inherent observable properties.

Expanding the size of datagrams can have implications for protocol operation. Trivially, it reduces the space in the path maximum transfer unit (PMTU) [PMTUv4], [PMTUv6] available to the multiplexed meta-protocol.

More substantially, the addition of octets to a protocol datagram - especially at the start - can also mean that the apparent format of datagrams changes. If a multiplexed meta-protocol adds octets to the start of payloads, this reduces the degree to which the multiplexed meta-protocol can be identified correctly. Some middleboxes have been shown to observe and discriminate based on the content of flows, even when large parts of a flow is encrypted and therefore inaccessible to the middlebox. Octets added to the start of a datagram could cause middleboxes to apply different treatment to a protocol when it is multiplexed than when it is not.

4. No Expectation of Successful Multiplexing

A multiplexing scheme that relies on certain properties of a protocol cannot expect those properties to remain fixed as that protocol changes.

It's possible that the designers of a protocol will explicitly guarantee that certain properties won't change over time, such as the invariants defined in [QUIC-INVARIANTS], in which case a multiplexing scheme could be designed based on those guarantees.

Choosing a multiplexing scheme that relies on properties of a protocol remaining constant can either impose unwanted constraints on the design of a protocol that is selected for multiplexing, or it can cause some or all features of that protocol to be unusable in the multiplexed context when that protocol changes.

5. Security Considerations

The ability to use new protocol features can have a material impact on security if access to updated or added security-related features is prevented by an incompatibility with a chosen multiplexing scheme.

6. IANA Considerations

This document has no IANA actions.

7. Informative References

[CHECKSUM] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", RFC 1071, DOI 10.17487/RFC1071, September 1988, <https://www.rfc-editor.org/info/rfc1071>.

[CHECKSUM-FAIL] Stone, J. and C. Partridge, "When the CRC and TCP checksum disagree", Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication - SIGCOMM '00, DOI 10.1145/347059.347561, 2000.

[DTLS] 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>.

[DTLS13] Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-02 (work in progress), October 2017.

[PMTUv4] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[PMTUv6] 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>.

[QUIC-INVARIANTS]
Thomson, M., "Version-Independent Properties of QUIC", draft-thomson-quic-invariants-00 (work in progress), November 2017.

[RT-MUX] Petit-Huguenin, M. and G. Salgueiro, "Multiplexing Scheme Updates for Secure Real-time Transport Protocol (SRTP) Extension for Datagram Transport Layer Security (DTLS)", RFC 7983, DOI 10.17487/RFC7983, September 2016, <https://www.rfc-editor.org/info/rfc7983>.

[RTP] 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>.

[STUN] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <https://www.rfc-editor.org/info/rfc5389>.

[TLS] 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>.

[TURN] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <https://www.rfc-editor.org/info/rfc5766>.

[ZRTP] Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, DOI 10.17487/RFC6189, April 2011, <https://www.rfc-editor.org/info/rfc6189>.

Author's Address

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com