TSVWG
Internet Draft
Intended status: Experimental
Intended updates: TBD
Expires: September 2024

J. Touch
Independent Consultant

March 3, 2024

The UDP Authentication Option
draft-touch-tsvwg-udp-auth-opt-00.txt

Abstract

This document extends UDP by defining a framework for an authentication option.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at https://www.ietf.org/shadow.html

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 September 3, 2024.

Table of Contents

1. Introduction

TBD

This document currently contains a copy of the text from draft-ietf-tsvwg-udp-options, to be updated.

2. Conventions used in this document

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.

In this document, the characters ">>" preceding an indented line(s) indicates a statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the portions of this RFC covered by these key words.

3. Terminology

The following terminology is used in this document:

o TBD

4. Background

TBD

5. Authentication (AUTH)

The Authentication (AUTH, Kind=9) option is intended to allow UDP to provide a similar type of authentication as the TCP Authentication Option (TCP-AO) [RFC5925]. AUTH covers the UDP user data. AUTH supports NAT traversal in a similar manner as TCP-AO [RFC6978].

Figure 1 shows the UDP AUTH format, whose contents are identical to that of the TCP-AO option, with the addition of a 32-bit unsigned sequence number. The sequence number is used to differentiate otherwise identical datagrams for cryptographic purposes; it is intended to not repeat during the lifetime of a security association, but are otherwise meaningless (e.g., they can be monotonically increased except during rollover). Because AUTH sequence numbers are not coordinated and not reliably transmitted, in contrast to TCP, they cannot be used to derive session traffic keys. During an association, the one-byte KeyID and ReceiveNextKeyID (RNKID) fields serve the same purpose as for TCP-AO, allowing the active keys used in either direction to change in a coordinated manner.

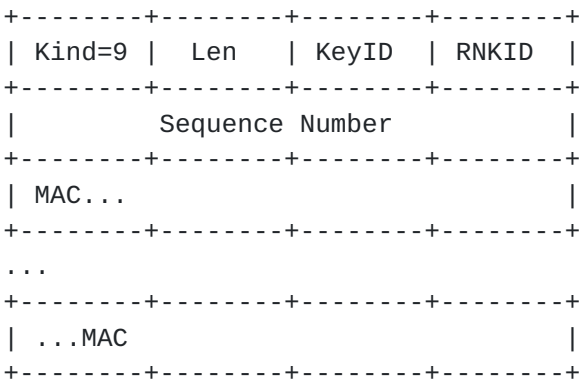

Figure 1 UDP AUTH option format

Like TCP-AO, AUTH is not negotiated in-band. Its use assumes both endpoints have populated Master Key Tuples (MKTs), used to exclude non-protected traffic.

TCP-AO generates unique traffic keys from a hash of TCP connection parameters. UDP lacks a three-way handshake to coordinate connection-specific values, such as TCP's Initial Sequence Numbers (ISNs) [RFC9293], thus AUTH's Key Derivation Function (KDF) uses

zeroes as the value for both ISNs. This means that the AUTH reuses keys when socket pairs are reused, unlike TCP-AO.

>> UDP packets with incorrect AUTH HMACs MUST be passed to the application by default, e.g., with a flag indicating AUTH failure.

>> UDP fragments with individual incorrect AUTH HMACs MUST be accumulated and passed to the application by default as part of the reassembled packet.

>> If used with UDP fragments, AUTH MUST be configured to cover the UDP option area (because fragments have an empty UDP data area).

Like all non-UNSAFE UDP options, AUTH needs to be silently ignored when failing. This silently-ignored behavior ensures that option-aware receivers operate the same as legacy receivers unless overridden.

In addition to the UDP user data (which is always included), AUTH can be configured to either include or exclude the surplus area (again, the latter is not allowed for UDP fragments), in a similar way as can TCP-AO can optionally exclude TCP options. When UDP options are covered, the OCS value and AUTH (and later, UENC) hash areas are zeroed before computing the AUTH hash. It is important to consider that options not yet defined might yield unpredictable results if not confirmed as supported, e.g., if they were to contain other hashes or checksums that depend on the surplus area contents. This is why such dependencies are not permitted except as defined for the OCS and the AUTH (and later, UENC) option.

Similar to TCP-AO-NAT, AUTH (and later, UENC) can be configured to support NAT traversal, excluding (by zeroing out) one or both of the UDP ports and corresponding IP addresses [RFC6978].

6. Security Considerations

TBD

UDP options are not covered by DTLS (datagram transport-layer security). Despite the name, neither TLS [RFC8446] (transport layer security, for TCP) nor DTLS [RFC9147] (TLS for UDP) protect the transport layer. Both operate as a shim layer solely on the user data of transport packets, protecting only their contents. Just as TLS does not protect the TCP header or its options, DTLS does not protect the UDP header or the new options introduced by this document. Transport security is provided in TCP by the TCP Authentication Option (TCP-AO [RFC5925]) or in UDP by the

Authentication (AUTH) option (Section 5) and UNSAFE Encryption (UENC) option (Section Error! Reference source not found.). T ransport headers are also protected as payload when using IP security (IPsec) [RFC4301].

7. IANA Considerations

TBD

8. References

8.1. Normative References

[Fa23] Fairhurst, G., T. Jones, "Datagram PLPMTUD for UDP Options," draft-ietf-tsvwg-udp-options-dplpmtud, Jun. 2023.

[RFC768] Postel, J., "User Datagram Protocol," RFC 768, August 1980.

[RFC791] Postel, J., "Internet Protocol," RFC 791, Sept. 1981.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts -- Communication Layers," RFC 1122, Oct. 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[RFC5925] Touch, J., A. Mankin, R. Bonica, "The TCP Authentication Option," RFC 5925, June 2010.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words," RFC 2119, May 2017.

8.2. Informative References

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, Dec. 2005.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field," RFC 6864, Feb. 2013.

[RFC6978] Touch, J., "A TCP Authentication Option Extension for NAT Traversal", RFC 6978, July 2013.

[RFC8126] Cotton, M., B. Leiba, T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs," RFC 8126, June 2017.

[RFC8200] Deering, S., R. Hinden, "Internet Protocol Version 6 (IPv6) Specification," RFC 8200, Jul. 2017.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018.

[RFC8504] Chown, T., J. Loughney, T. Winters, "IPv6 Node Requirements," RFC 8504, Jan. 2019.

[RFC9147] Rescorla, E., H. Tschofenig, N. Modadugu, "Datagram Transport Layer Security Version 1.3," RFC 9147, Apr. 2022.

[RFC9187] Touch, J., "Sequence Number Extension for Windowed Protocols," RFC 9187, Jan. 2022.

[RFC9293] Eddy, W. (Ed.), "Transmission Control Protocol," STD 7, RFC 9293, Aug. 2022.

[CERT18] CERT Coordination Center, "TCP implementations vulnerable to Denial of Service,", Vulnerability Note VU 962459, Software Engineering Institute, CMU, 2018, https://www.kb.cert.org/vuls/id/962459.

[To18] Touch, J., "A TCP Authentication Option Extension for Payload Encryption," draft-touch-tcp-ao-encrypt, Jul. 2018.

9. Acknowledgments

This work benefitted from discussions on the IETF TSVWG and SPUD email lists.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Joe Touch
Manhattan Beach, CA 90266 USA

Phone: +1 (310) 560-0334
Email: touch@strayalpha.com

Appendix A.Implementation Information

The following information is provided to encourage interoperable API implementations.

System-level variables (sysctl):

Name	default	meaning
net.ipv4.udp_opt	0	UDP options available
net.ipv4.udp_opt_ocs	1	Default use OCS
net.ipv4.udp_opt_apc	0	Default include APC
net.ipv4.udp_opt_frag	0	Default fragment
net.ipv4.udp_opt_mds	0	Default include MDS
net.ipv4.udp_opt_mrds	0	Default include MRDS
net.ipv4.udp_opt_req	0	Default include REQ
net.ipv4.udp_opt_resp	0	Default include RES
net.ipv4.udp_opt_time	0	Default include TIME
net.ipv4.udp_opt_auth	0	Default include AUTH
net.ipv4.udp_opt_exp	0	Default include EXP
net.ipv4.udp_opt_uenc	0	Default include UENC
net.ipv4.udp_opt_uexp	0	Default include UEXP

Socket options (sockopt), cached for outgoing datagrams:

Name	meaning
UDP_OPT	Enable UDP options (at all)
UDP_OPT_OCS	Use UDP OCS
UDP_OPT_APC	Enable UDP APC option
UDP_OPT_FRAG	Enable UDP fragmentation
UDP OPT MDS	Enable UDP MDS option
UDP OPT MRDS	Enable UDP MRDS option
UDP OPT REQ	Enable UDP REQ option
UDP OPT RES	Enable UDP RES option
UDP_OPT_TIME	Enable UDP TIME option
UDP OPT AUTH	Enable UDP AUTH option
UDP OPT EXP	Enable UDP EXP option
UDP_OPT_UENC	Enable UDP UENC option
UDP OPT UEXP	Enable UDP UEXP option

Send/sendto parameters:

Connection parameters (per-socketpair cached state, part UCB):

Name	Initial value
opts_enabled	net.ipv4.udp_opt
ocs_enabled	net.ipv4.udp_opt_ocs

>> The JUNK option is included for debugging purposes, and MUST NOT be enabled otherwise.

System variables

net.ipv4.udp_opt_junk 0

System-level variables (sysctl):

Name	default	meaning
net.ipv4.udp_opt_junk	0	Default use of junk

Socket options (sockopt):

Name	params	meaning
UDP_JUNK	-	Enable UDP junk option
UDP_JUNK_VAL	fillval	Value to use as junk fill
UDP_JUNK_LEN	length	Length of junk payload in bytes

Connection parameters (per-socketpair cached state, part UCB):

Name	Initial value
junk_enabled	net.ipv4.udp_opt_junk
junk_value	0xABCD
junk_len	4