Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: November 28, 2016

C. Paasch
Apple, Inc.
A. Ford
Pexip

May 27, 2016

TLS Authentication for MPTCP
draft-paasch-mptcp-tls-authentication-00

Abstract

Multipath TCP (MPTCP), described in [4], is an extension to TCP to provide the ability to simultaneously use multiple paths between peers.

draft-paasch-mptcp-application-authentication specifies "application layer authentication" for Multipath TCP, an alternatively negotiated keying mechanism for MPTCP. This allows keying material to be sourced from an application layer protocol in order to secure MP_JOIN handshakes.

This document explains how to use the proposed application-layer authentication extension with TLS [6], in order to leverage securely exchanged keys for MPTCP security, whilst simultaneously freeing the MPTCP token to be used as a channel for additional information.

Status of This Memo

Table of Contents

1. Introduction

As described in draft-paasch-mptcp-application-authentication, the use of "application-layer authentication" allows the Key used in MPTCP authentication to be provided by the application layer, thus permitting the use of existing secure communication channels for exchanging keying material. Furthermore, this decouples the key from the token and thus allows the token to be used for conveying additional semantics, such as helping front-end proxies route traffic to appropriate back-end servers.

TLS [6] provides a secure authentication channel between end hosts, where keys are not transmitted in the clear. The protocol generates a master secret for a connection, and a method is described in [3] for exporting a key generated from this and other properties which can then be used by the application layer. This document shows how to use this exported key, along with the method in draft-paasch-mptcp-application-authentication, for providing alternative keying mechanisms for MPTCP.

2. Technical Implementation

As described in draft-paasch-mptcp-application-authentication, the initial MP_CAPABLE handshake will exchange an arbitrary token for identifying an MPTCP connection. Whilst it is RECOMMENDED that the token is hard to guess, it can be used to carry any data, such as arbitrary routing information, and the security provided by the application-layer security will mitigate any risks of an attacker guessing tokens.

When an MPTCP end host wishes to open a new subflow, it will follow the same exchange as described in [4], however the keying material (Key-A and Key-B) will be derived from the TLS handshake, as described in [3]. The "label" field MUST be "EXPORTER-MPTCP". The length used in the key-derivation, following [3] is 16. Key-A are the 64 most-significant bits, while Key-B are the 64 remaining bits. This requires the key exchange to have completed before subflows can be created. Other than the source of the keys, the exchange remains the same. The MP_CAPABLE and MP_JOIN exchange therefore looks like this:

```
              Host A                                  Host B
     ------------------------                       ----------
     Address A1    Address A2                       Address B1
     ----------    ----------                       ----------
         |             |                                |
         |             |  SYN + MP_CAPABLE              |
         |--------------------------------------------->|
         |<---------------------------------------------|
         |         SYN/ACK + MP_CAPABLE(Token-B)        |
         |             |                                |
         |      ACK + MP_CAPABLE(Token-A, Token-B)      |
         |--------------------------------------------->|
         |             |                                |
         |             |   SYN + MP_JOIN(Token-B, R-A)  |
         |             |------------------------------->|
         |             |<-------------------------------|
         |             | SYN/ACK + MP_JOIN(HMAC-B, R-B) |
         |             |                                |
         |             |     ACK + MP_JOIN(HMAC-A)      |
         |             |------------------------------->|
         |             |<-------------------------------|
         |             |             ACK                |

   HMAC-A = HMAC(Key=(Key-A+Key-B), Msg=(R-A+R-B))
   HMAC-B = HMAC(Key=(Key-B+Key-A), Msg=(R-B+R-A))
```

Figure 1: Example Use of MPTCP Authentication

3. Security Considerations

This draft relies on the security provided by TLS [6] and the key export mechanism of [3] to provide additional security for the MPTCP handshake mechanism. These changes remove lingering risks, originally identified in [7], where an intercept of the initial MPTCP handshake could allow session hijack.

4. IANA Considerations

IANA would be requested to add a value to the TLS Exporter Label registry as described in [3]. The label is "EXPORTER-MPTCP".

5. References

5.1. Normative References

[1] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

[3] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <http://www.rfc-editor.org/info/rfc5705>.

[4] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", draft-ietf-mptcp-rfc6824bis-05 (work in progress), January 2016.

[5] National Institute of Science and Technology, "Secure Hash Standard", Federal Information Processing Standard (FIPS) 180-3, October 2008, <http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf>.

5.2. Informative References

[6] 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>.

[7] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, January 2013.

Authors' Addresses

Christoph Paasch
Apple, Inc.
Cupertino
US

EMail: cpaasch@apple.com

Alan Ford
Pexip

EMail: alan.ford@gmail.com