Use of Fingerprints for Identifying Certificates in the Session Description Protocol (SDP)
draft-thomson-mmusic-fingerprint-00

Abstract

The Session Description Protocol (SDP) fingerprint attribute binds a session description to an X.509 certificate. This document describes how hash agility is achieved without backwards compatibility issues. This document also describes how the fingerprint attribute can be used to identify a set of valid certificates.

This document updates RFC4572.

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 http://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 April 25, 2014.

Copyright Notice

Copyright (c) 2013 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 (http://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
1. Introduction

RFC 4572 [RFC4572] describes how the fingerprint Session Description Protocol (SDP) [RFC4566] attribute binds a session description to an X.509 certificate [X509]. Unfortunately, the only statement it makes regarding multiple fingerprints is the following:

A certificate fingerprint MUST be computed using the same one-way hash function as is used in the certificate’s signature algorithm.

This has the unfortunate consequence of unnecessarily coupling the security properties of the certificate to the hash function used in signing the certificate. To maximize the chances of a certificate being accepted, the hash algorithm used in certificates tends to lag current best practice, potentially exposing sessions to attacks based on hash collision.

The ability to use stronger cryptographic hash algorithms (hash agility) improves the integrity of the binding between the session description and the entity in possession of the private key. Systems that rely on other bindings to the session (or the private keys used to establish it) do not benefit from these changes.

This document also describes an optional mechanism that might be used to identify multiple certificates, in cases where the offered certificate might be selected from a small set. This might have operational advantages where the endpoint answering a call is not known ahead of a session description being created.
1.1. Conventions and Terminology

At times, this document falls back on shorthands for establishing interoperability requirements on implementations: the capitalized words "MUST", "SHOULD" and "MAY". These terms are defined in [RFC2119].

2. Hash Agility

The SDP fingerprint attribute is used to indicate the hash of the certificate - a certificate fingerprint - that is offered in the TLS [RFC5246] or DTLS [RFC5764] handshake. The certificate fingerprint so included is used to bind the (D)TLS session to the session description.

Multiple fingerprint attributes can be used to identify a certificate using alternative cryptographic hash algorithms. This allows sessions descriptions to use alternative, potentially stronger, hash algorithms without risking interoperability failure. A stronger hash algorithm is more resistant to collision attacks, which can be used to impersonate endpoints.

To avoid cases where certificate fingerprints cannot be validated, implementations MUST support SHA-256 [FIPS2]. That is, a fingerprint attribute using SHA-256 MUST be included in any place that includes fingerprint attributes and implementations MUST be able to validate SHA-256 fingerprints.

Implementations or specific applications can specify that validation using a different hash algorithm be mandatory in order to achieve a desired level of collision resistance. Implementations MUST NOT consider a session binding to be valid unless a certificate fingerprint using sufficiently strong hash algorithm matches one in the session description. For example, an application might specify that certificates are validated using SHA-384 [FIPS2] in addition to, or instead of, SHA-256.

Additional fingerprint attributes MAY be included using alternative hash algorithms. An endpoint that validates a session using fingerprint attributes MUST report failure if any hash that it checks doesn’t match.

Endpoints can ignore fingerprint attributes that use hash algorithms it doesn’t support or wish to validate.

3. Multiple Certificates
It might be that an application desires the ability to create session descriptions where the security context can be terminated using one of a small set of certificates.

An endpoint that validates a session description with multiple values for the same hash algorithm MUST fail the validation unless a fingerprint matches for each hash algorithm validated. Therefore, a session description that includes multiple a=fingerprint values for the same hash algorithm MUST include the same number of a=fingerprint values for every hash algorithm that is included.

4. Security Considerations

Over time, advances in cryptanalysis and computational power render hash algorithms increasingly prone to collision attacks. A hash collision on a certificate fingerprint would allow for impersonation. This document describes how to use different hash algorithms, independent of those selected for use in certificates.

Hash agility does not reduce the need for session description integrity protection or any of the suggested supporting mechanisms described in [RFC4572].

Adding support for hash agility does not affect the properties gained through the use of other mechanisms like the use of other bindings between session and identity. This document only improves the binding between a session and its description in SDP. For example, mechanisms such as the one described in [I-D.ietf-rtcweb-security-arch] require the implementation of equivalent processing rules to benefit from hash agility. Systems relying on the X.509 certificate chain to a specific trust anchor are similarly unaffected.

5. IANA Considerations

This document has no IANA actions.

6. Acknowledgements

Kevin Dempsey raised the original question that motivated this draft.

7. References
7.1. Normative References


7.2. Informative References


Author’s Address

Martin Thomson
Microsoft
3210 Porter Drive
Palo Alto, CA  94304
US

Email: martin.thomson@skype.net