Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 18, 2016

D. Zhang
D. Liu
Alibaba Group
March 17, 2016

A TLS Extension for Service Indication
draft-zhang-tls-service-indication-extension-00

Abstract

This memo specifies a service indication extension, which is used to identify the services or contents that a client is trying to access. This extension can be used in the scenarios such as reverse charging.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 September 12, 2016.

Copyright Notice

Table of Contents

1. Introduction

To attract potential consumers and gain advantages in the market competition, more and more ICPs seek to provide customers with discount for their traffics accessing their services. In order to achieve this, a ICP need to cooperate with its ISPs and enable the charging gateways of ISPs to distinguish the traffic flows accessing to certain content/services from other traffics and then charge them with different policies. The discount rate could be various for different customers, or for the same customers at different time or in different areas. In order to achieve this objective, additional information needs to be provided for a charging gateway so that the gateway can find the associated charging policies for the traffic flow. Such information should not be be provided at the application layer when TLS or other transporting layer security protocols have been widely used in practice. Otherwise, such information may be encrypted and un-reachable to the charging gateway.

This document specifies a TLS extension which carries the Service Indication (SI) informaiton. The extension is transferred in the first message from the client in the TLS handshake. Actually, the service name indication extension (SNI for short) can be potentially used to transfer such information. However, such SNI is not cryptographically protected, and anyone can generate a fake SNI and transfer it in the handshake to deceive the charging gateway. For instance, a user can transfer a service indication informaiton provided of ICP-A when it is accessing the service provided by ICP-B in order to gain more discounts in traffic fee. To avoid this

problem, the service indication information itself needs to be protected so that the gateway it is .

In the mechanism proposed in this memo, it is assumed the charging gateway and the customer APP shares an identical key, which is deployed by the ICPs in advance. However the way that the keys are actually deployed is out of scope. Such a key is used to generate a MAC for the SI information and a timestamp (which is used to prove the freshness of the information). The information and the MAC are both transferred in the extension. When receiving the a Client hello message from a customer, the charging gateway will honor the SI information only when the attached timestamp and the MAC are both valid.

2. Service Identification Extension

In order to indicate which service/content that a client intends to access, clients MUST include an extension of type "service_indicaiton" in the (extended) client hello. The 'ExtensionType' field of this extension contains "ServiceID(TBD)" The "extension_data" field of this extension SHALL contain "ServiceIndicatingInfo" where:

```
struct {

opaque ServieName:<1..2^16-1>;

uint64 timestamp;

KeyID key_identifier;

opaque Message_authenticaiton_data <1..2^64-1>;

} ServiceIndicatingInfo;

enum {

key_id(0), (2^16-1)

} KeyID;
```

key_identifer indicates the key and the associated hash algorithm used to generate message authentication code.

"timestamp" is the current NTP Time [RFC5905], measured since the epoch (January 1, 1970, 00:00), ignoring leap seconds, in milliseconds.In order to check the Timestamp field, recipients SHOULD be configured with an allowed timestamp Delta value, a "fuzz factor"

for comparisons, and an allowed clock drift parameter. The recommended default value for the allowed Delta is 300 seconds (5 minutes); for fuzz factor 1 second; and for clock drift, 0.01 second.

Before sending out the first client_hello packet, the client needs to fill the SNI, the timestamp, the key ID, and the authentication data into the extension. The way of generating the authentication data is introduced in Section 2.1. Upon receiving the client_hello, the gateway will check whether the timestamp is valid. Then it will also generate the authentication data and compare it with the authentication data transferred in the extension. If the result is false, the gateway will charge the traffic according to its local policies.

2.1. Generation of Authentication Data

In the algorithm description below, the following nomenclature, which is consistent with [FIPS-198], is used.

H is the specific hashing algorithm (e.g. SHA-256).

Ko is the cryptographic key used with the hash algorithm. As mentioned before, this key is pre-deployed.

B is the block size of H, measured in octets rather than bits. Note, that B is the internal block size, not the hash size. For SHA-1 and SHA-256 B is equal to 64. For SHA-384 and SHA-512 B is equal to 128. L is the length of the hash, measured in octets rather than bits.

XOR is the exclusive-or operation.

Opad is the hexadecimal value 0x5c repeated B times.

Ipad is the hexadecimal value 0x36 repeated B times.

Apad is the hexadecimal value 0x878FE1F3 repeated (L/4) times.

2.1.1. Preparation of the Key

In this application, Ko is always L octets long.

If the Authentication Key (K) is L octets long, then Ko is equal to K. If the Authentication Key (K) is more than L octets long, then Ko is set to H(K). If the Authentication Key (K) is less than L octets long, then Ko is set to the Authentication Key (K) with zeros appended to the end of the Authentication Key (K) such that Ko is L octets long.

2.1.2. First Hash

First, the Message Authentication Data field in the Extension is filled with the value of Apad . Then, a first hash, also known as the inner hash, is computed as follows:

First-Hash = H(Ko XOR Ipad || (Extension))

2.1.3. Second Hash

Then a second hash, also known as the outer hash, is computed as follows:

Second-Hash = H(Ko XOR Opad || First-Hash)

The second hash becomes the message authentication data and set into the message authentication data field of the extension. The length of the field is identical to the message digest size of the specific hash function H that is being used.

3. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

4. Security Considerations

This mechanism uses timestamps to address the replay attack issues. This mechanism is based on the assumption that the client and the charging gateway have roughly synchronized clocks, with certain allowed clock drift. So, accurate clock is not necessary. If one has a clock too far from the current time, the timestamp mechanism would not work.

For the consideration of overhead imposed to the charging gateway, this mechanism only consider the use of SHA-1 and SHA-256. In the future version of this memo, SHA384 and SHA512 could be considered according to the comments.

5. Acknowledgements

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5909] Combes, J-M., Krishnan, S., and G. Daley, "Securing Neighbor Discovery Proxy: Problem Statement", RFC 5909, DOI 10.17487/RFC5909, July 2010, <http://www.rfc-editor.org/info/rfc5909>.

Authors' Addresses

Dacheng Zhang
Alibaba Group

Email: dacheng.zdc@alibaba-inc.com

Dapeng Liu
Alibaba Group

Email: max.ldp@alibaba-inc.com