Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 22, 2016

D. Migault
K. Ma
Ericsson

January 19, 2016

TLS/DTLS Content Provider Edge Server Split Use Case
draft-mglt-lurk-tls-use-cases-00

Abstract

TLS as been designed to setup and authenticate transport layer between endpoints.

A lot of applications are using TLS in order to set communications between the applications end points.

As long as applications end points and transport end points were combined into the same host, application authentication could be combined with the transport authentication.

As the current internet is decoupling the transport and application layers, such model may not be applicable anymore. In other words, TLS authentication cannot be handled on behalf of the application authentication.

This document describes use cases where the authentication of the transport layer differs from the authentication performed at the application layer.

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 July 22, 2016.

Table of Contents

1. Introduction

TLS has been designed for end-to-end security between a TLS Server and a TLS Client. As TLS is widely used to provide an authenticated channel between applications, the following models assumes that applications end points and connectivity end point are combined. In that case, authentication of the connection end point and authentication of the application end point could be combined and assimilated as a single authentication.

Such assumption for the TLS model may not be true especially in the current web architecture where application content is not anymore associated with the connection end point. For example, Content Delivery Network are in charge of delivering content they are not necessarily owning.

This document provides use case where authentication of of the TLS Server involves multiple parties or entities as opposed to a single

entity in the standard TLS model. Such uses cases are designated a split use cases to point out that authentication is split between multiple entities.

2. Terminology

TLS Client: The TLS Client designates the initiator of the TLS session. The terminology is the one of [RFC5246]. The current document considers that the TLS Client and the application initiating the session are hosted on the same host. If not they are hosted on the same administrative domain with a trust relation between the TLS Client and the application. In other words, the client endpoint is considered to be a single entity as described initially in [RFC5246].

TLS Server: The TLS Server designates the endpoint of a TLS session initiated by the TLS Client. This document considers that application end points and the TLS session end point my be hosted on different nodes, and may belong to different administrative domains.

Edge Server: The Edge Server designates a node that handles traffic for a Content Provider. A TLS Client initiates a TLS session to authenticate a Content provider, but may be in fact served by a Edge Server that may belong to a different administrative domain.

Content Provider: The owner of the content. This is the entity requested by the application of the TLS Client.

Content Delivery Network (CDN): designates a organization in charge of managing delivery of a content on behalf of a Content Provider. In most cases, the CDN is a different organization than the Content Provider.

3. Cloud Use Case

It is common that applications - like a web browser for example - use TLS to authenticate a Content Provider designated by a web URL and build a secure channel with that Content Provider.

TLS provides end-to-end security between the two end points of the communication. In our case, the two end points are the web browser also designated as TLS Client and the other end point is the TLS Server hosting the content. When the TLS Server is the Content Provider, the web browser or the TLS Client can use TLS to set an authenticated and secure channel between the web browser and the Content Provider using TLS. On the other hand, TLS can hardly be

used in a secure, scalable way when the TLS Server differs from the Content Provider.

Suppose that the content of the Content Provider cannot be hosted by a single server. This may be the case for example when a single server is not anymore sufficient to address all the load of the TLS Clients. In this case, the load may be split between multiple instance of servers. Similarly, a Content Provider may chose to place multiple instance of the servers with a limited subset of the content at different places in the network in order to avoid traffic to be conveyed through the whole data center or the content provider infrastructure. In most of the cases, these instances of the servers are places at the edge of the infrastructure. Another reason for having multiple instances may be that the content provider cannot host the entirety of its content on one single server. In that case it may opt for hosting the content on various servers to which the application can be directly connected.

When the Content Provider cannot present a single server, the web applications can connect to, it clearly appears that the Content Provider the application is trying to set an authenticated TLS session with and the TLS end point may differ. In the latter case, the TLS end points are designated as Edge Servers. These Edge Servers are used for connectivity and should operate transparently for the application. In other words, the application requires that authentication of the application layer be performed at the transport layer.

In order to enable the web application to authenticate the Content Provider using TLS, two options may be considered:

a): Each Edge Server shares the authentication credential associated to the Content Provider. This case results in each Edge Server usurping the identity of the Content Provider.

b): The authentication credentials of the Content Provider are kept secret, and any TLS session between a web application and an Edge Server involves some interaction between the Edge Server and the Content Provider.

Section 5 evaluates these two possibilities.

4. Content Delivery Network Use Case

The Content Delivery Network Use case is similar as the Cloud use case exposed in Section 3. The main difference is that the Edge Servers are not anymore under the responsibility of the Content provider. Instead, the Content Provider has subscribed to a company

with a different administrative domain to manage the Edge Server on behalf of the Content Provider.

In the case of Content Distribution Network Interconnection (CDNI) [RFC6707], it may also that the company with which the Content Provider has contracted may further delegate delivery to another CDN with which the Content Provider has no official business relationship. Even if the Content Provider trusts the upstream CDN, and perhaps has strong legal contracts in place, it has no control over, and possibly no legal recourse against, the further downstream CDNs.

The same options as in Section 3 apply, but in case of sharing authentication credential of the Content Provider, these credentials are shared outside the administrative borders of the Content Provider.

5. Authentication in Split Scenarios

Access by the Edge Server to the private or secret information of the Content Provider may be performed either by sharing the information between the Content Provider and the Edge Servers or by using an interface between the Edge Servers and the Content Provider.

Sharing secret information between the Content Provider and the Edge Servers increases the risk of leaking information. The risk of leaking information exists as Edge Servers are exposed on the Internet which present a high surface of potential attack as illustrated for example by the Heartbleed attack [HEART]. More specifically, the Heartbleed attack uses a weakness of a software implementation to retrieve the private key used by the TLS server. Such attack would not for example has been so successful if the private key was not stored on the Edge Server.

In addition, the risk increases with the number of Edge Servers and the number of organizations sharing these secrets. In fact when the Edge Servers increases and are managed by multiple organizations, it becomes hard for the Content Provider to control the conformance of the Edge Servers to the security policies enforced by the Content Provider, or to detect when a leakage occurs. At last, from a security point of view, this may be not acceptable the Content Provider .

Currently an interface between the Edge Server and the Content Provider has not been standardized. This may prevent a Content Provider to interact with multiple CDNs, or multiple CDNs to interoperate. In addition, such interface may be used within a CDN in order to manage its own Edge Servers. The absence of a standard

interface and the lack of interoperability may also result in the Content Provider sharing the confidential information with a third party organization. This may be not acceptable in a security point of view.

6. Security Considerations

One motivation for split scenario is to avoid spreading authentication credentials of the Content Provider in multiple Edge Servers, and so to reduce the leak of such credentials.

On the other hand, preventing the authentication credentials to be hosted on the Edge Servers do not necessarily prevent any leakage. In fact, the Edge Servers and the Content Providers are likely to use a specific channel that provide access to the credentials. It is of primary importance to design this channel to avoid the Content Provider to reveal any information about the private key. More specifically, even though a Edge Server may become corrupted or under the control of an attacker, the attacker should not be able to be able to disclose the authentication credentials.

7. IANA Considerations

There are no IANA considerations in this document.

8. Acknowledgements

Thanks are due for insightful feedback on this document to Robert Skog, Salvatore Loreto, John Mattson and Rich Salz.

9. References

9.1. Normative References

[RFC5246] 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>.

[RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, DOI 10.17487/RFC6707, September 2012, <http://www.rfc-editor.org/info/rfc6707>.

9.2. Informative References

[HEART] Codenomicon, "The Heartbleed Bug", <http://heartbleed.com/>.

Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Kevin Ma J
Ericsson
43 Nagog Park
Acton, MA 01720
USA

Phone: +1 978-844-5100
Email: kevin.j.ma@ericsson.com