OpenSSL for Tomcat

Initial plan

Supervised by Jean-Frederic Clere (Redhat)

Project coordinator: Dr. Hugues Mercier

Co-instructor: Pr. Jacques Savoy

TEAM AND CONTACT INFORMATION:

Numa de Montmollin (numa.demontmollin@unifr.ch)

John Hannay (john.hannay@unifr.ch)

Table of contents

OpenSSL for Tomcat 1
Table of contents 2
Version 3
Project Description 4
Context 4

Version

The version of this paper is changed only if a major change take place. Otherwise, only a decimal is changed (1.0 to 1.1)

version	Update date	Page modified
1.0	22.03.2015	1-9

Project Description

OpenSSL for Tomcat is a try to connect directly OpenSSL to Tomcat. It means that the TLS/SSL encryption in Tomcat will be handled by OpenSSL, a well-know fast and stable open-source implementation of the TSL/SSL protocol. The connection will be done through the NIO2 Java API and will call OpenSSL with the Java Native Interface (JNI).

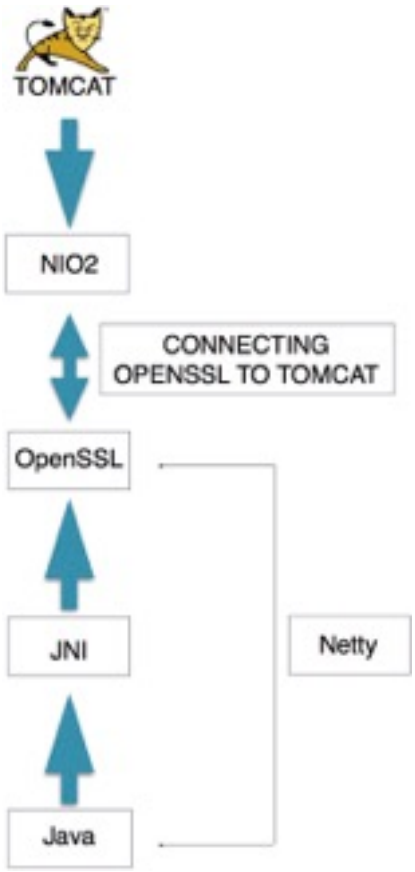

FIGURE 1: PROJECT OVERVIEW

The figure 1 schematize how the project looks like. Basically, OpenSSL is binded into Java through the JNI. This have been already done in the Netty project, an event-driven networking application framework. The connection will be to call OpenSSL by using NIO2 and in consequence to give to Tomcat a new way to realize the TSL/SSL encryption.

Tomcat is an open-source implementation of the Java EE specifications for the Java Servlet Container and the Java Server Page (JSP). It implements as well a HTTP connector and a Web server.

Context

For large-scale Web applications, TSL/SSL encryption is a critical part in terms of performance and security. It can consume a lot of CPU time and can slow-down the entire application. Thus, the implementation of the protocol can be buggy and relay to critical securities issues. For example, the Heartbleed vulnerability which affected many big companies like Facebook and Twitter.

With Apache Tomcat, there exist three possibilities of using TSL/SSL encryption:

1. The default option, ships with Tomcat, is using Java Secure Socket Extension (JSSE). The Java implementation of TLS/SSL protocol.

2. With Tomcat Native and using OpenSSL for TLS/SSL through the Apache Portable Runtime (APR)
3. With Tomcat native for Netty and using OpenSSL for TLS/SSL provided by Netty.

Three of them are relying problems:

1. JSSE has a lack of performance comparing to other implementations of TLS/SSL as shown in figure 1
2. Tomcat native use APR which has a lot of C code. In consequence it is hard to maintain and probably contains some bugs.
3. Tomcat native for Netty requires a lot of dependencies

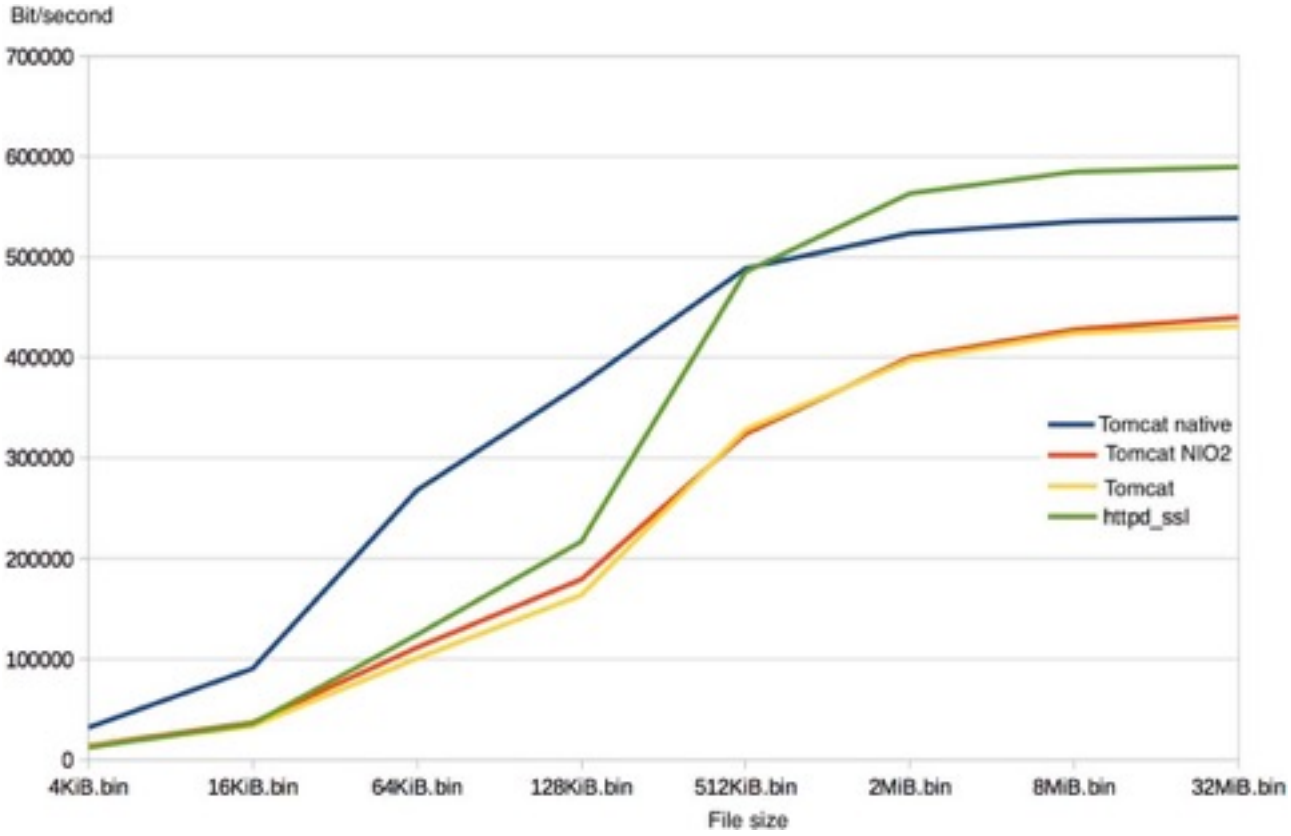

FIGURE 1: PERFORMANCE ON DIFFERENT IMPLEMENTATIONS OF TSL/SSL

Our client is Redhat, a company which is active in developing open-source solutions. More precisely we are working with Jean-Frederic Clere, employee of Redhat Neuchâtel and Tomcat committer.