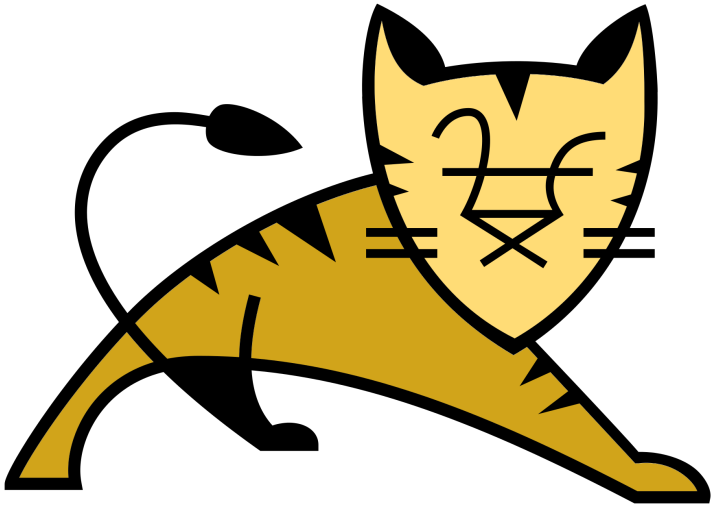
OpenSSL for Tomcat

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

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

Our project OpenSSL for Tomcat aims to connect directly OpenSSL to Tomcat. This 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).

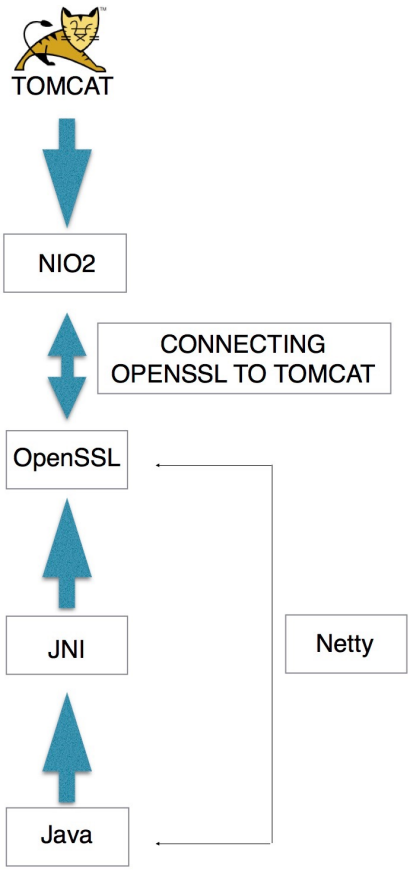


FIGURE 1: PROJECT OVERVIEW

Figure 1 graphically describes what the project looks like. Basically, OpenSSL is bound into Java through the JNI. This has already been done in the Netty project, an event-driven networking application framework. The connection between NIO2 and OpenSSL will enable OpenSSL and NIO2 to work together, and as a result would give Tomcat a new way to realize the TSL/SSL encryption.

Context

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

Tomcat native is an optional component for Tomcat which increase the operating system compatibility and gives access to sockets and OpenSSL for TLS/SL through the Apache Portable Runtime (APR). APR is a collection of software libraries with a predictable API based on platform specific softwares.

Tomcat native for Netty is a fork of Tomcat native which aims to remove APR and use the Netty framework. Especially for TLS/SSL encryption which is provided by Netty with the OpenSSL library.

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. The implementation of the protocol can be buggy and relay to critical securities issues. One such example is the Heartbleed vulnerability[1] 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 with Tomcat uses Java Secure Socket Extension (JSSE). The Java implementation of TLS/SSL protocol
2. With Tomcat Native and using OpenSSL for TLS/SSL through the APR
3. With Tomcat native for Netty and using OpenSSL for TLS/SSL provided by Netty

Each one of them has their drawbacks:

1. JSSE lacks performance in comparison to other implementations of TLS/SSL as shown in figure 2
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

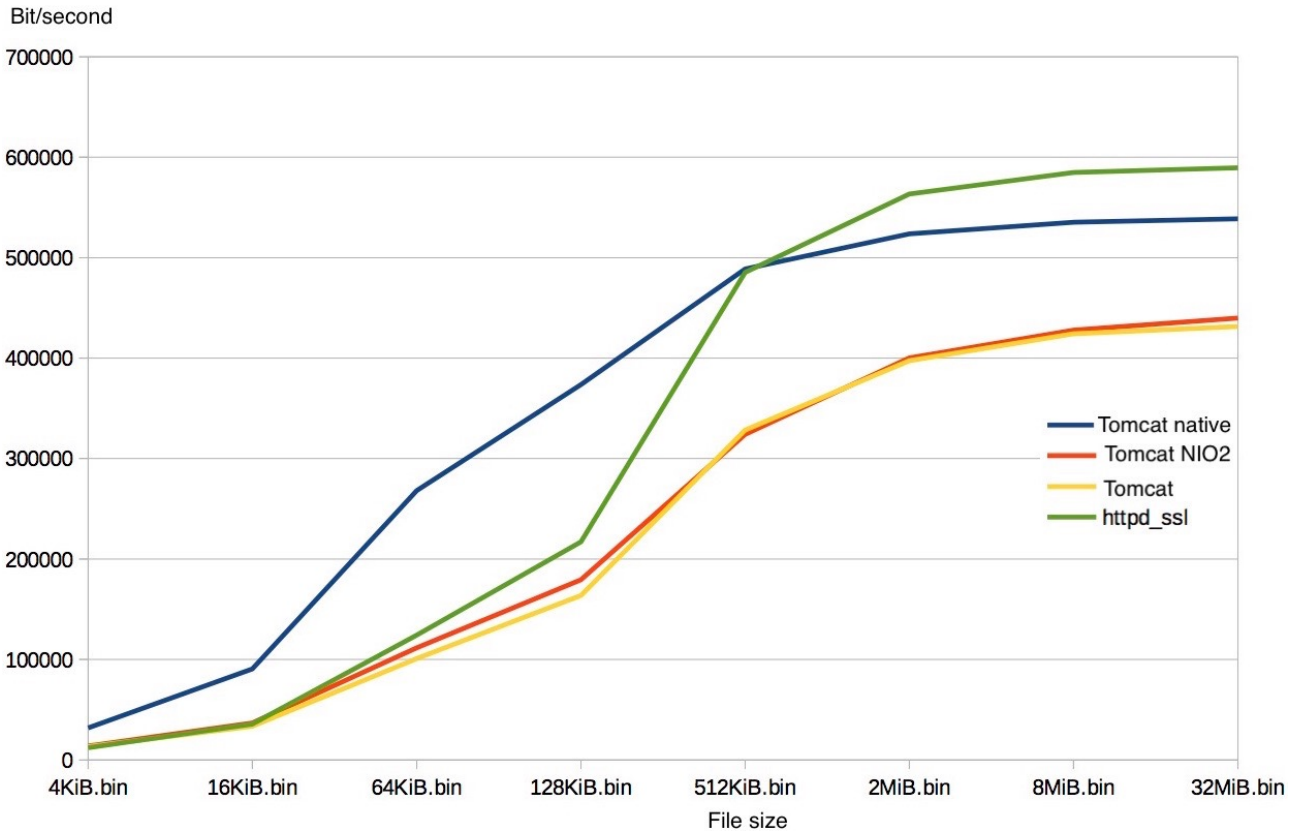


FIGURE 2: PERFORMANCE ON DIFFERENT IMPLEMENTATIONS OF TSL/SSL[2]

Our client is Redhat, a provider of open-source solutions. It is one of the leading company in this domain and is famous for selling a slightly modified version of Linux. More precisely,

[1] https://www.us-cert.gov/ncas/alerts/TA14-098A and http://heartbleed.com

[2] Extracted from a presentation of Jean-Frederic Clere and modified for our needs

we are working with Jean-Frederic Clere, employee of Redhat Neuchâtel and Tomcat committer.

Goals and objectives

The main goal of this project is to provide an open-source integration of OpenSSL into Tomcat. The implementation should also be faster than Tomcat with JSSE. More precisely, this means we need :

- extract the required code from Netty
- Build a connection from OpenSSL (extracted from Netty) to NIO2
- Based on Tomcat native, provide an optional component for Tomcat for using OpenSSL for TLS/SSL encryption
- As this project is open-source, a good documentation must be provided.
- At the end of the project benchmarks will be run and the results should be better than Tomcat using JSSE for the TSL/SSL encryption

An optional goal would be to acquire external contributors to the project and hoping for a continued development after our project ceases.

Project organization

This projects requires that we find and read a lot of documentation and code. Moreover, we will probably have a lot of code to maintain and we will need to write a report and a presentation as well. In consequence, we have decided to set up two Git repository on Github.

Tomcat-openssl-doc (https://github.com/facenord-sud/tomcat-openssl-doc) is for the documentation related to the project, this involves: presentations, reports, work-plans, etc. We use the Github wiki of this project to share information between each other. For example, when one of us finds out how to import Tomcat under eclipse and how to build it (easy), he write an entry in the wiki. Proceeding like this save times and there will already be some texts written for the final report.

Tomcat-openssl (https://github.com/facenord-sud/tomcat-openssl) is for our implementation. This involves, the code we write it and the code we take from other projects. We plan to use this repository for the documentation directly considering the code we write. For example, the Javadoc, how to build our project, how to use it, etc. For assigning special task to each other and signalling bugs, we have plan to use the Github system issues of this project. This permits us to have a structured way to discuss and to keep traces.

Responsibility distribution

Unlike most projects, it is difficult to share responsibilities. This is due to the fact that the project will probably not require much coding but to understand the large projects (Tomcat, OpenSSL, Tomcat native, Tomcat native for Netty) and then to extract and adapt the required code to solve our problem.

This demands that both of us are active and understand each individual technology. We have however decided that when we encounter some problem to solve that we will split these evenly. Assigning responsibility will be in case by case for a specific problem. To help us do this, we will use the Github issues system which allows us to assign an issue to a person and to discuss it together.

Interactions with the client

Jean-Frederic Clere has asked us to write a logbook each week to see the progress in the project and to keep control on what is going on.

Jean-Frederic Clere is willing to answer our questions regarding Apache Tomcat and OpenSSL, we intend to ask him questions when stuck in a specific domain during the project.

We have already been to Redhat twice to meet with Jean-Frederic Clere to prepare the project. No periodic meeting are scheduled but we can come when we want to work and ask questions.

Methodology

For this project we do not have do special choices on which languages, tools or libraries to use. Everything is already decided and contained, we will code in Java, and our code should integrate into Tomcat. We plan to extract a large amount of code from Netty and probably from Tomcat native as well. Tomcat native for Netty could also be a source of inspiration.

At this point, it is hard to say exactly how we will proceed because we have not yet acquired all the necessary knowledge. The general method is to build Tomcat, Tomcat native and Netty by ourselves and after that to start doing modifications. Proceeding like this gives us the possibility that if something fails, we know that before a modification everything was going well and is easier to investigate.

As development environment we use Netbeans, Vim or IntelijIdea Community Edition (depending of the project) for reading and writing the code. We use Git as the source version control.

All the tools we are using have open-source licenses. All the projects we plan to use code from are licensed under the apache 2 software license which is a really permissive license for reusing the code. Our work will be open-source as well. Which license to use and if we need one is not yet determined.

State of the art

Our project is not related to research papers, but a lot of documentation is bundled with repositories of the projects Tomcat (http://apache.tomcat.org), Tomcat native (https://github.com/tomcat/tomcat-native) and Netty (https://netty.io). The commented code and mailing lists are another source of information.

Constraints and elements of risk

RISK ONE:

- Understanding large projects such as Tomcat, OpenSSL, Tomcat native and Tomcat native for Netty has a potential of risk. We could spend hours still not understanding the project or to be lost in the code, Netty is about a hundred thousands lines of code.

However, we believe that we have assigned sufficient time in our work-plan to achieve our goals due to the vast amount of documentation on the projects. On top of that Jean-Frederic Cere has offered to answer our questions if we have difficulties in understanding certain parts of the projects. Because these are open-source projects, it is always possible to ask questions (and to expect an answer) from the community through IRC or the mailing lists.

RISK TWO:

- Many similar open-source projects were abandoned which could indicate an element of risk.

This is true, but these projects used NIO as opposed to NIO2. According to our client, NIO2 should make life easier for us.

RSK THREE:

- The biggest risk in our opinion is getting lost or branching out of our objectives by getting stuck in technical details which don't advance our project. This could lead to a potential of time wasting.

In order to limit such risk, we propose to follow our work-plan as closely as possible. We will also write a weekly progress report, this should allow us to recognize potential time wasting.

Deliverable goods

We will deliver an implementation of using OpenSSL for TLS/SSL encryption directly into Tomcat.

These goods will be delivered on Github in the repository facenord-sud/tomcat-openssl (https://github.com/facenord-sud/tomcat-openssl-doc), this should encourage the open-source community to extend and maintain our project. In order to make our project more attractive to the community, we will provide it with extensive documentation.

Efforts and schedule

Task	estimated Hours	Finished by
set git up	1	—
search for documentation and code	1	—
study doc + code	4	—
prepare presentation 1	4	—
Initial plan	3	—
DEADLINE 1: 22.03.2015	**13**	**22.03**
create a dummy server with SSLEngine of Java	5	27.03
understand how ssl is used in Tomcat	10	3.04
understand the differences with tomcat-native	10	10.04
implement the same dummy server but while using OpenSSl	20	24.04
DEADLINE 2: 30.04.2015	**45**	**30.04**
integrate Openssl in tomcat (probably in a similar way of as tomcat-native)	25	8.05
write the report	30	26.05
final presentation	20	5.06
DEADLINE 3: ~ 10.06.2015	**75**	**~ 10.06**
TOTAL	**133**	

This wok plan has been extracted from the Github wiki of the repository facenord-sud/tomcat-openssl-doc (https://github.com/facenord-sud/tomcat-openssl-doc/wiki/Workplan). It may change over time.

Credits

The Tomcat logo from the first page and the figure 1 is from http://commons.wikimedia.org/wiki/File:Tomcat-logo.svg Consulted 22.03.2015

Thanks to Jean-Frederic Clere for the graphic of the figure 2.