Overview

''Ralf Engelschall has released an excellent module that integrates Apache and SSLeay.''

Tim J. Hudson, SSLeay F.A.Q.

Global Table Of Contents

Chapter 1: Preface
Chapter 2: Introduction
Chapter 3: Reference
Chapter 4: Compatibility
Chapter 5: HowTo
Chapter 6: F.A.Q. List
Chapter 7: Glossary

This module provides strong cryptography for the Apache (v1.3) webserver via the Secure Socket Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols by the help of the excellent SSL/TLS implementation library OpenSSL from Eric A. Young and Tim Hudson.

The mod_ssl package was created in April 1998 by Ralf S. Engelschall and was originally derived from the Apache-SSL package developed by Ben Laurie. It stays under a BSD-style license which is equivalent to the license used by The Apache Group for the Apache webserver itself. This means, in short, that you are free to use it both for commercial and non-commercial purposes as long as you retain the authors' copyright notices and give the proper credit.

Legalese

Although the above conditions also apply to Apache and OpenSSL in general (both are freely available and useable software packages), you should be aware that especially the cryptographic algorithms used inside OpenSSL stay under certain patents and perhaps import/export/use restrictions in some countries of the world. So whether you can actually use the combination Apache+mod_ssl+OpenSSL in your country depends mainly on your local state laws. The authors of neither Apache nor mod_ssl nor OpenSSL are liable for any violations you make here.

If you're not sure what law details apply to your country you're strongly advised to first determine them by consulting an attorney before using this module. A lot of hints you can find in the International Law Crypto Survey which is a really comprehensive resource on this topic. At least two countries with heavy cryptography restrictions are well known: In the United States (USA) it's not allowed to (re-)export mod_ssl or OpenSSL And inside France it's not allowed to use any cryptography at all when keys with more than 40 bits are used.

> This software package uses strong cryptography, so while it is created, maintained and distributed from Germany and Switzerland (where it is legal to do this), it falls under certain export/import and/or use restrictions in some other parts of the world.
>
> PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. SO, WHEN YOU IMPORT THIS PACKAGE TO YOUR COUNTRY, RE-DISTRIBUTE IT FROM THERE OR EVEN JUST EMAIL TECHNICAL SUGGESTIONS OR EVEN SOURCE PATCHES TO THE AUTHOR OR OTHER PEOPLE YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHOR OF MOD_SSL IS NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFULLY YOURSELF, IT IS YOUR RESPONSIBILITY.
>
> CREDIT INFORMATION: This product includes software developed by Ben Laurie for use in the Apache-SSL HTTP server project, software developed by Larry Wall and David MacKenzie for use in the GNU project of the FSF and software developed by Dr. Stephen N. Henson as a companion to OpenSSL.

Module Architecture

The mod_ssl package consists of the SSL module (part 1 in Figure 1) and a set of source patches for Apache adding the Extended API (EAPI) (part 2 in Figure 1) which is an essential prerequisite in order to use mod_ssl. In other words: you can only use the mod_ssl module when Apache's core code contains the Extended API. But because when applying mod_ssl to the Apache source tree the Extended API is also automatically added you usually don't have to think about this. It's mainly important for package vendors who want to build separate packages for Apache and mod_ssl. For more details on how to apply mod_ssl to the Apache source tree please follow the `INSTALL` file in the mod_ssl distribution.

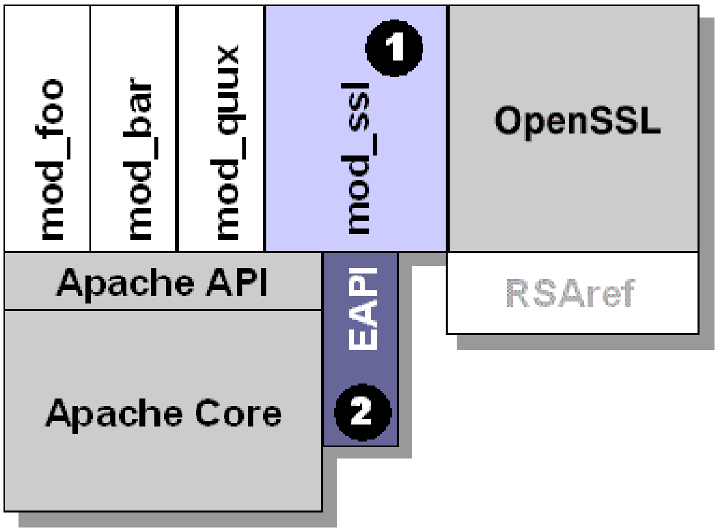

Figure 1: Module Architecture

Module Building

The SSL module (mod_ssl) resides under the `src/modules/ssl/` subdirectory inside the Apache source tree and is a regular Apache module. This means that you can configure, build and install it like any other Apache module. Usually this is done by using the APACI command

```
$ cd apache_1.3.x/
$ SSL_BASE=/path/to/openssl ./configure ... --enable-module=ssl
```

or by manually editing the `SSL_BASE` variable, uncommenting the corresponding `AddModule` directive inside the `src/Configuration` file and using the command

```
$ cd apache_1.3.x/src
$ ./Configure
```

for configuring. Additionally you can enable the Dynamic Shared Object (DSO) support for mod_ssl by either adding the `--enable-shared=ssl` option to the APACI configure command line or by replacing the

```
AddModule ssl_module modules/ssl/libssl.a
```

line in `src/Configuration` with

```
SharedModule ssl_module modules/ssl/libssl.so
```

Building mod_ssl as a DSO is especially interesting to achieve more run-time flexibility, i.e. you can decide whether to use SSL or not at run-time instead of build-time. But notice that building mod_ssl as a DSO requires that your OS/compiler supports building DSOs in the first place, and additionally that they support linking of a DSO against a static library (libssl.a, libcrypo.a). Not all platform support this.