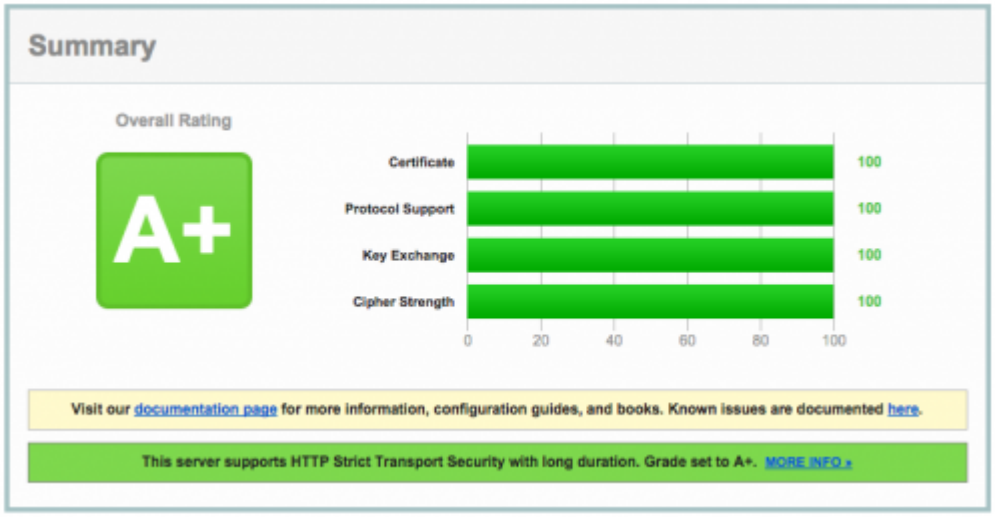
Table of Contents

Good HTTPS with Apache HTTPD 2.2+

Introduction

This document aims to achieve PFS in HTTPS connections using Apache HTTPD 2.2+ and OpenSSL 1.0.1+. It does not cover general Apache HTTPD or OpenSSL installation and configuration and thus is pretty much straight forward. The term 'good' was measured using the Qualys SSL Labs Test Page, achieving the highest possible rating as pictured below:



The SSL certificate was obtained for free from StartSSL. Of course, older browsers and operating systems (and unfortunately, all Java versions including Java 8, hahahaha) get locked out. For the HPKP implementation (see below) to be standards compliant, you'd need at least 2 certificates - the second one for example from CACert.

NOTE: *The StartSSL CA certificate is included in most (if not all) browsers, however for CAcert, this is not the case. This document explains how to import CA certificates under different OS/Browser combinations.*

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. The author shall have no liability for any error or damages of any kind resulting from the use of this document. There is no warranty; not even for merchantability or fitness for a particular purpose.

Required Apache modules

You'll need to have the `mod_headers` and obviously the `mod_ssl` Apache modules installed and working.

Suggested config changes

Protocols

For now, we can only assume TLSv1.2 to be not totally broken:

```
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

Cipher Suites

We only want ephemeral Diffie-Hellman ciphers, and we want them in our particular order:

```
SSLCipherSuite ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-
SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA:ECDHE-ECDSA-AES256-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK
SSLHonorCipherOrder on
```

HSTS

HSTS refers to “HTTP strict transport security” and involves setting a special HTTP header in your Apache config:

```
Header add Strict-Transport-Security "max-age=15768000; includeSubDomains"
```

HPKP

HPKP refers to “HTTP Public Key Pinning” and involves 2 steps.

Calculating checksums

For ease of use, the checkums were obtained using this tool. Instructions on doing this by hand can be found here.

Setting header

Having obtained the `Public-Key-Pins` HTTP header from the tool mentioned before, this header is added to the Apache config:

```
Header set Public-Key-Pins "pin-sha256=\"checksum-of-cert-1\"; pin-
sha256=\"checksum-of-cert-2\"; max-age=15768000; includeSubDomains"
```

Further considerations

DANE, https://ssl-tools.net/tlsa-generator.
https://www.owasp.org/index.php/HTTP_Strict_Transport_Security for more Info about HSTS and strict HSTS.

crypto, openssl, https, apache, hsts, hpkp, pfs