Network Working Group W. Hoehlhubmer
Internet-Draft Nov 7, 2013
Category: Best Current Practice
Intended status: Standards Track
Expires: May 7, 2014

Informational Add-on for HTTP over the Secure Sockets Layer (SSL) Protocol and/or the Transport Layer Security (TLS) Protocol

draft-hoehlhubmer-https-upd-04

Abstract

This document describes an Add-on as a good practice for websites providing encrypted connectivity (HTTP over TLS).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Encrypted connections are not limited to just one way doing this.

A list of a few encryption algorithms:

(1) Advanced Encryption Standard (AES)
(2) Data Encryption Standard (DES, 3DES)
(3) Ron's Code 4 (RC4)
(4) ...

As an example a list of some kinds of the Camellia encryption algorithm [CAMELLIA] (names taken from OpenSSL help [OPENSSL]):

(1) CAMELLIA-128-CBC: 128-bit Camellia encryption in CBC mode
(2) CAMELLIA-128-ECB: 128-bit Camellia encryption in ECB mode
(3) CAMELLIA-192-CBC: 192-bit Camellia encryption in CBC mode
(4) CAMELLIA-192-ECB: 192-bit Camellia encryption in ECB mode
(5) CAMELLIA-256-CBC: 256-bit Camellia encryption in CBC mode
(6) CAMELLIA-256-ECB: 256-bit Camellia encryption in ECB mode

A list of possible secure layer used:

(1) The Secure Sockets Layer (SSL) Protocol:
  (1a) Version 2.0 [SSLv2]
  (1b) Version 3.0 [SSLv3]

(2) The Transport Layer Security (TLS) Protocol:
  (2a) Version 1.0 [TLSv1.0]
  (2b) Version 1.1 [TLSv1.1]
  (2c) Version 1.2 [TLSv1.2]

A list of possible Ciphersuites for Transport Layer Security (TLS):

(1) Pre-Shared Key Cipher Suites [RFC4279]
(2) Elliptic Curve Cryptography (ECC) Cipher Suites [RFC4492]
(3) Pre-Shared Key (PSK) CipherSuites with NULL Encryption [RFC4785]
(4) AES Galois Counter Mode (GCM) Cipher Suites [RFC5288]
(5) DES and IDEA Cipher Suites [RFC5469]
(6) ECDHE_PSK Cipher Suites [RFC5489]
(7) Camellia Cipher Suites [RFC5932]
(8) ...

A list of a few hashing algorithms:

(1) the MD5 Message-Digest Algorithm [MD5] used commonly in past
(2) the US Secure Hash Algorithm 1 (SHA1) [SHA1] used in present
(3) ...

Only the X.509 Certificates [PKIX] are static, all other informations depend on the capabilities of the used web browser.

Because not every browser allows you to view all these informations, especially the Cipher Suite the browser has picked for use, it is a good practice to show these informations on the website.

With most browsers you can view the used X.509 certificates of the actual session, but have no direct comparison if they are the correct ones. The X.509 certificates which are shown by the browser and these from this Add-on MUST match; other ways there is going on a man-in-the-middle attack.

2. Implementing this Add-on

This Add-on is just one page of the website. Its content MUST be completely generated on server side. The Common Gateway Interface [CGI] is RECOMMENDED to be used. There MUST be a relative reference to this page as defined in [RFC3986] Section 4.2.

For doing so see the sample scripts at Appendix A.

To see how this Add-on works, see [MYADDON].

2.1. Content of this Add-on

The informations MUST be the following:

(1) The actual date and time formatted as specified in [RFC5322] Section 3.3. It MUST NOT differ more than 5 seconds from actual date/time

(2) The cipher specification name

(3) Number of cipher bits (actually used)
(4) Number of cipher bits (possible)

(5) The SSL Protocol version: SSLv2, SSLv3, TLSv1.0, TLSv1.1, TLSv1.2, ...

(6) If cipher is an export cipher: false, true
(7) If secure renegotiation is supported: false, true

(8) Algorithm used for the public key of server's certificate
(9) Algorithm used for the signature of server's certificate
(10) Issuer DN of server's certificate
(11) Subject DN in server's certificate
(12) The serial of the server certificate
(13) The version of the server certificate
(14) Validity of server's certificate (start time)
(15) Validity of server's certificate (end time)

(16) Client certificate verification: NONE, SUCCESS, GENEROUS or FAILED:reason

(17) SSL compression method negotiated: NULL when disabled

For connections where X.509 certificates are used for authentication these informations are RECOMMENDED:

(18) Algorithm used for the public key of client's certificate
(19) Algorithm used for the signature of client's certificate
(20) Issuer DN of client's certificate
(21) Subject DN in client's certificate
(22) The serial of the client certificate
(23) The version of the client certificate
(24) Validity of client's certificate (start time)
(25) Validity of client's certificate (end time)
(26) Number of days until client's certificate expires

This information MAY be given:

(27) The hex-encoded SSL session id
(28) Contents of the SNI TLS extension (if supplied with ClientHello)

These OPTIONAL informations depend on the used software:

(29) The SSL-module program version: e.g. Apache mod_ssl version
(30) The SSL program version: e.g. OpenSSL version

See Appendix B for a sample content.

2.2. Formating/Presenting of this Add-on

You SHALL present this information simple, plain Text is enough. When using HTML there MUST NOT be any linked in resources from 3rd party and using only a subset of [HTML2.0] is RECOMMENDED. The content SHOULD NOT be translated to any other language. Presenting the content in sorted order is OPTIONAL.

3. IANA Considerations

There are no requests for IANA actions in this document.

4. Security Considerations

When implementing this information as a popup window in the browser, this information MUST also be available with enabled popup-blocker.

The Implementation MUST NOT use any scripts, that run on client side: e.g. Javascript, ...

There SHOULD also be no references to other websites inside this Add-on page.

5. Acknowledgements

6. Recommendations

Using a standardized URL is RECOMMENDED, for more see Section 8.

7. References

7.1. Normative References

7.2. Informative References

[CAMELLIA] Matsui, M., Nakajima, J., and S. Moriai, "A Description of the Camellia Encryption Algorithm", RFC 3713, April 2004.

[CGI]   Robinson, D. and K. Coar, "The Common Gateway Interface (CGI) Version 1.1", RFC 3875, October 2004.

[HTML2.0]   Berners-Lee, T. and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, November 1995.

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[PKIX] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[SHA1] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

[SSLv2] Hickman, Kipp, "The SSL Protocol", Netscape Communications Corp., Feb 9, 1995.

[SSLv3] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, August 2011.

[TLS1.0] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[TLS1.1] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[TLS1.2] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[OPENSSL] OpenSSL Cryptography and SSL/TLS Toolkit at http://www.openssl.org/

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4279] Eronen, P., Ed., and H. Tschofenig, Ed., "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4492] Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, May 2006.

[RFC4785] Blumenthal, U. and P. Goel, "Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)", RFC 4785, January 2007.

[RFC5288]   Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, August 2008.

[RFC5322]   Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC5469]   Eronen, P., Ed., "DES and IDEA Cipher Suites for Transport Layer Security (TLS)", RFC 5469, February 2009.

[RFC5489]   Badra, M. and I. Hajjeh, "ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)", RFC 5489, March 2009.

[RFC5932]   Kato, A., Kanda, M., and S. Kanno, "Camellia Cipher Suites for TLS", RFC 5932, June 2010.

[MYADDON]   A working implementation of this Add-on on my website at https://ssl.mathemainzel.info/sslinfo/

8. Discussions

It MAY be good to have a standardized URL for this Add-on; e.g. https://www.example.com/sslinfo/

A releative reference at the main page inside the encrypted part is RECOMMENDED.

It is OPTIONAL to place an Absolute URI as defined in [RFC3986] Section 4.3. outside the encrypted website part.

Appendix A. Script Examples

Use the following script examples as a template for your implementation of this Add-on.

The first two examples generate identical content, the third example presents the content of Section 2.1. unsorted.

(a) Example 1. CGI-script, can be used on most Linux systems
(b) Example 2. PHP-script
(c) Example 3. CGI-script (a compiled C program)

Example 1. CGI-script, can be used on most Linux systems

```
<CODE BEGINS>
#!/bin/sh

printf "Content-type: text/plain\n\n"

printf "SSL informations: $(date --rfc-2822)\n"
printf "================\n\n"

if [ "$HTTPS" == "on" ]; then
  env | grep --regexp="^SSL_" | sort
else
  printf "No SSL information available.\n"
fi
<CODE ENDS>
```

Example 2. PHP-script

```
<CODE BEGINS>
<?php

header( "Content-type: text/plain" );

print "SSL informations: " . date( "r" ) . "\r\n";
print "================\r\n\r\n";

if ( isset( $_SERVER['HTTPS'] ) && ( $_SERVER['HTTPS'] == "on" ) ) {
  $list = array( );
  $nmbrOfValues = 0;
  foreach ( $_SERVER as $key => $value ) {
    if ( substr( $key, 0, 4 ) == "SSL_" ) {
      $list[ $nmbrOfValues++ ] = $key . "=" . $value;
    }
  }
  sort( $list );   // sort content before printing ...
  for ( $iter = 0; $iter < $nmbrOfValues; $iter++ ) {
    print $list[ $iter ] . "\r\n";
  }
}
else {
  echo "No SSL information available.\r\n";
}
?>
<CODE ENDS>
```

Example 3. CGI-script (a compiled C program)

```
<CODE BEGINS>
/* Compiles with GNU C compiler on Linux, Windows, ...
 *
 * When using Microsoft C/C++ in Windows, strftime format specifiers
 *   for timezone behave in a non-standard way;
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#ifdef __linux__
#include <unistd.h>
#endif

int main( int argc, char* argv[ ], char** envp )
{                              /* char* envp[ ] */
  char* psz;
  char szDateTime[ 80 ];

  time_t tnow = time( NULL );
  struct tm* tmnow = localtime( &tnow );

  strftime( szDateTime, sizeof( szDateTime ) - 4,
    "%a, %d %b %Y %H:%M:%S %z", tmnow );

  printf( "Content-type: text/plain\r\n\r\n" );

  printf( "SSL informations: %s\r\n", szDateTime );
  printf( "================\r\n\r\n" );

  if ( ( psz = getenv( "HTTPS" ) ) && ( strcmp( psz, "on" ) == 0 ) )
  {
    char** ppsz = envp;

    /* print content without sorting ... */
    while ( ppsz && *ppsz )
    {
      if ( strncmp( *ppsz, "SSL_", 4 ) == 0 )
        printf( "%s\r\n", *ppsz );
      ppsz++;
    }
  }
  else
    printf( "No SSL information available.\r\n" );

  return 0;
}
<CODE ENDS>
```

Appendix B. Add-on Sample Content

The first example shows a complete Add-on sample content in sorted order, the other examples shows only that may differ when using another browser.

For meaning of the numbers in brackets see Section 2.1.

(a) Example 1. A complete sample content
(b) Example 2.
(c) Example 3.

Example 1. A complete sample content

```
SSL informations: Thu, 01 Jan 1970 00:00:00 +0000              (1)
================

SSL_CIPHER=AES256-SHA                                          (2)
SSL_CIPHER_ALGKEYSIZE=256                                      (4)
SSL_CIPHER_EXPORT=false                                        (6)
SSL_CIPHER_USEKEYSIZE=256                                      (3)
SSL_CLIENT_VERIFY=NONE                                         (16)
SSL_COMPRESS_METHOD=NULL                                       (17)
SSL_PROTOCOL=TLSv1                                             (5)
SSL_SECURE_RENEG=true                                          (7)
SSL_SERVER_A_KEY=rsaEncryption                                 (8)
SSL_SERVER_A_SIG=sha1WithRSAEncryption                         (9)
SSL_SERVER_I_DN=/C=--/O=SomeOrg/OU=SomeOrgUnit/CN=Root CA      (10)
SSL_SERVER_I_DN_C=--                                           (10)
SSL_SERVER_I_DN_CN=Root CA                                     (10)
SSL_SERVER_I_DN_O=SomeOrg                                      (10)
SSL_SERVER_I_DN_OU=SomeOrgUnit                                 (10)
SSL_SERVER_M_SERIAL=01                                         (12)
SSL_SERVER_M_VERSION=3                                         (13)
SSL_SERVER_S_DN=/C=AT/CN=www.example.com                       (11)
SSL_SERVER_S_DN_C=AT                                           (11)
SSL_SERVER_S_DN_CN=www.example.com                             (11)
SSL_SERVER_V_END=Dec 31 23:59:59 1970 GMT                      (15)
SSL_SERVER_V_START=Jan 01 00:00:00 1970 GMT                    (14)
SSL_SESSION_ID=0000000000000000000000000000000000000000000     (27)
SSL_TLS_SNI=www.example.com                                    (28)
SSL_VERSION_INTERFACE=mod_ssl/2.2.15                           (29)
SSL_VERSION_LIBRARY=OpenSSL/1.0.0-fips                         (30)
```

Example 2.

```
SSL informations: Thu, 01 Jan 1970 00:00:00 +0000
================
...
SSL_CIPHER=RC4-MD5
SSL_CIPHER_ALGKEYSIZE=128
SSL_CIPHER_EXPORT=false
SSL_CIPHER_USEKEYSIZE=128
SSL_CLIENT_VERIFY=NONE
SSL_COMPRESS_METHOD=NULL
SSL_PROTOCOL=SSLv3
SSL_SECURE_RENEG=false
...
```

Example 3.

```
SSL informations: Thu, 01 Jan 1970 00:00:00 +0000
================
...
SSL_CIPHER=AES128-SHA256
SSL_CIPHER_ALGKEYSIZE=128
SSL_CIPHER_EXPORT=false
SSL_CIPHER_USEKEYSIZE=128
SSL_CLIENT_VERIFY=NONE
SSL_COMPRESS_METHOD=NULL
SSL_PROTOCOL=TLSv1.2
SSL_SECURE_RENEG=true
...
```

Author's Address

Walter Hoehlhubmer
Lederergasse 47a
A-4020 Linz
Austria, EUROPE

EMail: walter.h@mathemainzel.info