Informational Add-on for HTTP over
the Secure Sockets Layer (SSL) Protocol and/or
the Transport Layer Security (TLS) Protocol
draft-hoehlhubmer-https-upd-02

Abstract

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

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

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 7, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
1. Introduction

Encrypted connections are not limited to just one encryption algorithm. A few encryption algorithms, a short and incomplete complete list:

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

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

Examples for hashing algorithms are:

(1) the MD5 Message-Digest Algorithm [MD5] used commonly in past
(2) the US Secure Hash Algorithm 1 (SHA1) [SHA1] used in present
(3) ...
Interesting in this connection is the secure layer, too:

(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) ...

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 main page of 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 [CGI] MAY be used. For doing so see the sample scripts at Appendix A.

2.1. Formatting/Presenting of this Add-on

You SHALL present this information simple, plain Text is enough. The use of any extension to [HTML2.0] is NOT RECOMMENDED. The content SHOULD NOT be translated to any other language. Presenting the content in sorted order is OPTIONAL.
2.2. 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
These OPTIONAL informations depend on the used software:

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

See Appendix B for a sample content.

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

I recommend using a standardized URL, for more see Section 8.

7. References

7.1. Normative References


7.2. Informative References


8. Discussions

It MAY be good to have an standardized URL for this Add-on; e.g. https://www.example.com/sslinfo/
in case the word "ssl" is already part of the URL’s subdomain: e.g. ssl.example.com or www-ssl.example.com, then https://ssl.example.com/info/ or https://www-ssl.example.com/info are also fine for this standardized URL.

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.
Example 1. CGI-script, used on Linux systems

```sh
#!/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 -r-regexp="^SSL_" | sort
else
    printf "No SSL information available.\n"
fi
```

Example 2. PHP-script

```php
<?php

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

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

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

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

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

    return 0;
}
```

Appendix B. Add-on Sample Content

The first example shows a complete Add-on sample content generated by one of the scripts in Appendix A, the other example shows only that differs when using another browser. For meaning of the numbers in brackets see Section 2.1.

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=00000000000000000000000000000000000000000000000000000000000  (27)
SSL_VERSION_INTERFACE=mod_ssl/2.2.15 (28)
SSL_VERSION_LIBRARY=OpenSSL/1.0.0-fips (29)
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  
...  

Author’s Address

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

EMail: walter.h@mathemainzel.info