National Cyber Security Centre
Ministry of Security and Justice

HTTPS could be a lot more secure

Check configurations and apply new options

Factsheet FS-2014-03
version 1.1 | 5 November 2014

HTTPS is a frequently used protocol for protecting web traffic against parties setting out to eavesdrop on or manipulate the traffic. Configuring HTTPS requires precision: there are many options, and by no means all of them are secure.

This factsheet explains three HTTPS options that can contribute to securing web traffic. These options are additions to existing recommendations on the safe configuration of HTTPS. The NCSC recommends using these options in all of your HTTPS configurations.

The NCSC recommends protecting all websites that process sensitive data with HTTPS. If you want to protect your website with HTTPS you will find configuration advice in the IT security guidelines for Transport Layer Security and the IT security guidelines for web applications of the NCSC.

The following parties have contributed to this factsheet:
- Tax and Customs Administration
- DefCERT
- Schuberg Philis
- SIDN

The most important facts

- HTTPS is a protocol for securely transmitting web traffic between a browser and a web server. HTTPS protects the confidentiality and integrity of the data. It can also be used to identify the owner.
- The NCSC recommends protecting all websites that process sensitive data with HTTPS.
- In recent years some HTTPS options have gained popularity: HSTS, forward secrecy and SHA-2 for certificates. These options provide a higher level of protection. The NCSC recommends using these options for all websites that are protected with HTTPS.
- If you use HTTPS on your website, make sure that the configuration is secure. For that purpose apply the IT security guidelines for Transport Layer Security and the IT security guidelines for web applications of the NCSC.

The role of HTTPS

HTTPS is a protocol for securely transmitting web traffic between a browser and a web server. The traffic is sent via an internet connection or another non-trusted connection. HTTPS stands for 'HTTP Secure'. The protocol amounts to sending HTTP traffic via a connection that is secured with Transport Layer Security (TLS)[1]. The connection is protected with cryptography. The security of an HTTPS connection is guaranteed by means of one or two certificates.

HTTPS protects the confidentiality and integrity of the data when it is being transmitted. The confidentiality of data is important when sending sensitive data, such as personal data. It must not be possible for an outsider to view this data. The integrity of data is important when sending instructions, such as bank transfers. This must be authentic data that cannot be manipulated by outsiders.

HTTPS also provides facilities for identifying the party being communicated with. An Extended Validation (EV) certificate can be used to have a server confirm the identity of its owner. That way, a person who navigates to an internet banking site with an EV certificate knows not only that his details have been sent to the right website, but also that it really is the website of his bank. Some

[1] TLS is the new name for the protocol previously known as Secure Sockets Layer (SSL).

browsers support this message visually, such as by turning the address bar green. This increases a visitor's confidence in the website he has visited.

The Encrypt the Web initiative

In 2009 the Electronic Frontier Foundation announced the Encrypt the Web initiative[2]. The initiative calls on administrators to protect all their websites with HTTPS. The traffic transmitted to those websites will then be less vulnerable to interception or monitoring. This will also reduce the chance of administrators forgetting to configure a website containing sensitive data for HTTPS since they will be using this measure on all their websites.

Several large websites are now offering HTTPS. Well-known examples include Google, Facebook, Twitter and Wikipedia. Using the browser plugin HTTPS Everywhere[3] enables users to automatically be redirected to the HTTPS version of a website if one is available. The website must be registered in the plugin for this to work.

In August 2014 Google announced[4] that websites offering HTTPS would be placed higher in search results. Google is thus setting out to encourage the use of HTTPS.

What does the NCSC recommend?

The NCSC recommends protecting all websites that process sensitive data with HTTPS. Is the confidentiality of the data important, or its integrity? In that case, HTTPS is an appropriate measure, possibly with an EV certificate. For certain categories of websites this will almost always be the case:

- websites with forms in which visitors are asked to fill in person data;
- websites that process financial or medical information;
- websites for which users log in, e.g. with a username and password;
- websites that link to a website on which users have to log in;
- websites for which confidence in the identity of the owner is important (use EV certificates for this purpose).

If you use HTTPS on your website, make sure that the configuration is secure. The underlying security protocol, TLS, features many options. By no means all of them are secure. To select a secure TLS configuration for your website you can make use of the IT security guidelines for Transport Layer Security of the NCSC[5]. HTTPS also features some options that are exclusively for web traffic. These are covered in the IT security guidelines for web applications of the NCSC[6].

In recent years some HTTPS options have gained popularity: HSTS, forward secrecy and SHA-2 for certificates. These options provide a higher level of protection. The NCSC recommends using these options for all websites that are protected with HTTPS. These options are covered in this factsheet. They also form part of the IT security guidelines of the NCSC mentioned above.

HTTP Strict Transport Security (HSTS)

What HTTP Strict Transport Security (HSTS) is a technique used to instruct a browser only to visit a website via HTTPS. The web server sends an extra header that gives the browser this instruction. For each subsequent visit to this website for a set period the browser will directly ask for an HTTPS version of the website.

Why If a website uses HSTS a returning visitor will always visit the website via HTTPS. Attackers setting out to manipulate the network connection between the browser and the web server will thus be prevented from changing the connection from HTTPS back to HTTP. Attacks of this nature will however remain possible if a visitor goes to a website for the first time. HSTS does not provide any protection against this.

How This option must be activated on the web server to switch to HSTS. HSTS features two options:

- The web server administrator can configure the duration of the instruction only to visit the website via HTTPS. A period of six months to a year is commonly used. A longer period is beneficial because that way infrequent visitors are well-protected too. This does however involve sacrificing a degree of flexibility. If you want to stop offering your website via HTTPS in the future, you will have to wait until the duration of the HSTS instruction has expired in all visiting browsers.
- The web server administrator can decide whether the HSTS instruction also applies to subdomains. In that case a visit to the website example.com is sufficient to protect the website sub.example.com. You should only use this setting if you also intend to provide all websites on subdomains with HTTPS.

Instructions for configuring HSTS on your web server are available online. For Apache[7], IIS[8] and nginx[9] you will find references in the footnotes.

[2] See https://www.eff.org/encrypt-the-web.

[3] HTTPS Everywhere is available for several browsers. The plugin can be downloaded at https://www.eff.org/https-everywhere.

[4] See http://googleonlinesecurity.blogspot.nl/2014/08/https-as-ranking-signal_6.html.

[5] The IT security guidelines for Transport Layer Security are currently only available in Dutch: https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/whitepapers/ict-beveiligingsrichtlijnen-voor-transport-layer-security-tls.html. They will be available in English by the end of 2014.

[6] The IT security guidelines for web applications are only available in Dutch. See https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/whitepapers/ict-beveiligingsrichtlijnen-voor-webapplicaties.html.

[7] Instructions for Apache: https://www.owasp.org/index.php/HTTP_Strict_Transport_Security.

[8] HSTS is configured in IIS with a custom header: http://www.iis.net/configreference/system.webserver/httpprotocol/customheaders. The header to be configured is available on the page with the Apache instructions.

[9] Instructions for nginx: https://scotthelme.co.uk/setting-up-hsts-in-nginx/.

Forward secrecy

What Forward secrecy is a property of certain encryption methods for HTTPS. If you use an encryption method with forward secrecy the connection cannot be spied on by an attacker, even if he later obtains the secret key to the certificate used.

Why Forward secrecy is a significant element in the following scenario. If an attacker has access to the network between the browser and the web server he will be able to monitor secure HTTPS traffic. If he later succeeds in obtaining the secret key to the server certificate he will be able to decrypt the intercepted HTTPS traffic. He could obtain the secret key, for example, by hacking a computer or by means of legal proceedings, if necessary years later. Without forward secrecy it becomes increasingly important to protect the secret key of the server certificate because the confidentiality of ever more data retroactively depends on it.

How Activating forward secrecy amounts to using cipher suites for TLS with 'DHE' or 'ECDHE'[10] in their name. A cipher suite is a setting for encryption algorithms in TLS. Cipher suites are configured in the TLS configuration of the device where the TLS connection ends. In a simple setup that is the web server itself. In more complex environments this could also be the load balancer or an external provider of anti-DDoS measures.

When an HTTPS connection is established the server selects the cipher suite with its highest preference that is also supported by the browser. The server's preference is laid down in the TLS configuration. For that reason cipher suites with forward secrecy should be put at the top of this list so that they are selected if also supported by the client.

Instructions for configuring forward secrecy on various servers are available online. However since they are strongly entwined with recommendations on cipher suite selection, we have not provided any links here. See the IT security guidelines for Transport Layer Security for advice on the cipher suites to be configured. See the documentation for your TLS programming library for instructions on configuring cipher suites.

Certificates with SHA-2

What Certificate authorities place their digital signature on the hash, the digital fingerprint of the certificate. This makes it possible for a visitor to check the authenticity of the certificate. If an attacker succeeds in creating a counterfeit certificate with the same hash, he will be able to reuse the signature of the certificate authority on his counterfeit certificate.

The hash of a certificate is calculated with a hash function. A hash function is a mathematical function that 'garbles' data into a digital fingerprint. A secure hash function makes it virtually impossible to create two different certificates with the same hash.

Certificates with SHA-1 as the hash function are replaced by certificates with hash functions from the SHA-2 family: SHA-256, SHA-384 and SHA-512. Certificates with MD5 as the hash function were replaced some years ago. MD5 is the predecessor of SHA-1.

Why SHA-1 is an obsolete hash function. Browser vendors have announced that they will stop accepting SHA-1 certificates in the near future[11,12,13]. From the end of 2014 browsers will alert their users if websites present SHA-1 certificates that will remain valid after 2016. SHA1 certificates will no longer be accepted after 1 January 2017.

The reason for phasing out SHA-1 certificates is the risk of counterfeit certificates. A counterfeit certificate makes it possible for an attacker with network access to undermine the confidentiality and integrity of HTTPS connections. It has been demonstrated that this is possible in practice with another obsolete hash function, MD5[14]. It is to be expected that the same technology will soon also be applicable to SHA-1.

How Certificate authorities decide for themselves which hash function they use to sign a certificate. If you are currently using certificates based on SHA-1 (or even MD5), you should ask your certificate authority when you can replace them with copies based on hash functions from the SHA-2 family. Ask whether new certificates are only supplied on the basis of hash functions from the SHA-2 family.

The SHA-2 family of hash functions contains three functions that are regarded as forming a suitable basis for signing certificates: SHA-256, SHA-384 and SHA-512. It is stated in each certificate which hash function has been used to sign it.

To conclude

HTTPS is an important measure for protecting web traffic containing sensitive data. This is becoming increasingly common with modern websites. At the same time, securely configuring HTTPS takes precision. HSTS, forward secrecy and certificates based on SHA-2 are measures that improve the security of web traffic.

HTTPS is being used on ever more websites. That is understandable: it is an easily available measure that sends out a clear message that a website takes its users' privacy seriously. If you configure HTTPS properly, everyone who uses your websites will ultimately benefit.

[10] DHE stands for Diffie-Hellman Ephemeral. ECDHE stands for Elliptic Curve Diffie-Hellman Ephemeral. More background information about these cipher suites and this advice is given in the IT security guidelines for Transport Layer Security of the NCSC.

[11] Microsoft has announced that it will shortly stop accepting SHA-1 in Windows: http://blogs.technet.com/b/pki/archive/2013/11/12/sha1-deprecation-policy.aspx.

[12] Google has announced that it will shortly stop accepting SHA-1 in Google Chrome: http://googleonlinesecurity.blogspot.nl/2014/09/gradually-sunsetting-sha-1.html.

[13] Mozilla has announced that it will shortly stop accepting SHA-1 in Mozilla Firefox: https://blog.mozilla.org/security/2014/09/23/phasing-out-certificates-with-sha-1-based-signature-algorithms/.

[14] See http://www.win.tue.nl/hashclash/rogue-ca/.

How to proceed:

1. The NCSC recommends protecting all websites that process sensitive data with HTTPS. Also bear in mind how sensitive the details are for those they concern.
2. Make sure that HTTPS is properly configured. For that purpose apply the IT security guidelines for Transport Layer Security and the IT security guidelines for NCSC web applications.
   - » The use of HTTP Strict Transport Security (HSTS) forms part of the IT security guidelines for web applications.
   - » The use of forward secrecy forms part of the IT security guidelines for Transport Layer Security.
   - » The use of certificates with SHA-2 is an aspect of certificate management[15], but also arises in the IT security guidelines for Transport Layer Security.
3. Apply the selected HTTPS configuration to the web server and related devices such as load balancers.
4. Insights regarding which HTTPS options are secure change from time to time. This is in response to investigators discovering new attack methods, for example. Be sure to keep informed of these new insights and incorporate them in your HTTPS configuration. The NCSC guidelines mentioned above could be helpful in this regard.

[15] The factsheet Veilig beheer van digitale certificaten provides recommendations on good certificate administration (only available in Dutch): https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/factsheets/factsheet-veilig-beheer-van-digitale-certificaten.html.

Publication of **National Cyber Security Centre**
Turfmarkt 147 | 2511 DP The Hague | The Netherlands
P.O. Box 117 | 2501 CC The Hague | The Netherlands
www.ncsc.nl/english | info@ncsc.nl | T +31 70-751 55 55 | F +31 70-322 25 37
Publication No.: FS-2014-03 1.1 | No rights may be derived from this information.