Poster: Browser's "search form" issues and countermeasures

Yuji Suga suga@iij.ad.jp
Internet Initiative Japan Inc.,
Iidabashi Grand Bloom, 2-10-2 Fujimi, Chiyoda-ku, 102-0071, Japan

***Abstract*—From 2014, we are conducting fixed point observation to crawl SSL/TLS sites using .jp domain URL list extracted from Alexa Top Sites, and investigation on improvement of usage rate of SSL/TLS versions and Export-grade encryption algorithms. Furthermore, paying attention to the server side certificates, since the notation policy of the browser security indicator had recently changed, the green bar is displayed in the URL notation part originally although it uses the EV (Extended Validation) SSL certificate, it is "not safe" though sites that are judged were also found. As a situation similar to this issue, a detailed investigation was conducted on the browser's "search form" issues which are originally described to be safe although it is said to be unsafe due to inadequate site-contents. In this paper, the survey targeted are the websites of regular members belonging to the association which is planning and managing settlement systems and the like in "a certain" industry. We investigated SSL/TLS sites of Top FQDN which are widely announced on paper medium etc, so it was found that about half of them were in normal situation but half had problems such as FQDN mismatch. Moreover we also show the result of manually investigating the influence of the above "search form" issues by carrying out some pattern classification on the path reached from the HTTP (not HTTPS) server of the Top FQDN to the user login page. Finally, the design guideline of HTTP/HTTPS sites is mentioned as one of countermeasures against this kind of problems.**

Keywords–SSL/TLS; Extended Validation certificates

I. The History of SSL/TLS Version Changes

SSL 2.0 was released by Netscape Communications in 1995, and after a number of extensions were added and a number of issues fixed, SSL 3.0 was released the following year. SSL 2.0 had no function for preventing the alteration of the Handshake message portion (i.e. data integrity is not guaranteed), so MITM attacks were possible, and the protocol itself is recognized as vulnerable. Also, with the discovery of the POODLE attack in October 2014, padding oracle attacks against SSL 3.0 are now possible when the CBC cipher mode is used to encrypt messages, so it is currently recommended that SSL 3.0 not be used.

TLS, the successor to SSL, now has three versions: TLS 1.0 (established in 1999), TLS 1.1 (established in 2006), and TLS 1.2 (established in 2008). Each of these protocols are still in widespread use. After TLS 1.0 was drawn up by the IETF based on SSL3.0, TLS 1.1 was then designed to bolster its security by, for example, incorporating measures in its specifications beforehand to prevent the BEAST attack and its variants that the original protocol was vulnerable to when using CBC cipher mode. TLS 1.2 also enabled the use of authenticated encryption (AEAD: Authenticated Encryption with Associated Data). But these protocols have been targeted in many attacks over the past few years. RFC7457, which was issued in February 2015, summarizes the history of attacks against TLS that were known to the public by around 2014. It covers a wide variety of known attacks, pointing out vulnerabilities related to the RC4 stream cipher, which we will introduce next, and discussing downgrade attacks that force users to use a lower TLS version than expected, as well as timing attacks that occur when the compression function is enabled. For attacks after that period, portal sites such as CELLOS [1] can be checked for information on major SSL/TLS vulnerabilities, but it has become extremely difficult to accumulate knowledge about the respective attacks and deal with them each time they appear.

As an example, let's look at cases involving the RC4 and TripleDES cryptographic algorithms. RC4 is a well-known stream cipher that has been used extensively to date, and is defined in the cipher suites within SSL/TLS. A wide range of attack models can be considered when attacking cryptographic algorithms, but for cryptographic protocols like SSL/TLS, there is a condition requirement called Broadcast setting, when considering a real use case. This is an assumption where a large amount of ciphertext can be obtained from the same plaintext (data before being encrypted) that is encrypted using multiple keys. When considering how SSL/TLS is used, this is a fairly realistic use case. A large amount of research and cryptanalysis based on this attack model has been performed since 2001, and attacks where plaintext can be recovered through the bias of the stream keys generated by RC4 have been published. Specifically, a technique for generating large volumes of ciphertext by running malicious JavaScript in a browser has been written, and a paper presented at USENIX Security 2015 reported that it was possible to steal a cookie with a success rate of 94% by obtaining 9×2^{27} ciphertexts. In response to the various research and findings, the IETF considered this a real threat and issued RFC7465: in February 2015 to eliminate the use of RC4.

Meanwhile, the SWEET32 attack further reinforced the fact that TripleDES is vulnerable. This is not an attack method against a cryptographic algorithm itself, but a potential attack that could be successful when using the CBC cipher mode in SSL/TLS. This makes it impossible to prevent completely, and countermeasures by vendors all involve limiting or lowering the priority for use of TripleDES. Next, we will focus on the resources required to conduct this attack successfully. A paper presented at ACM CCS'16 stated that to restore a 2-block cookie would require capturing 785 gigabytes of ciphertext over a 38 hour period. RC4 bias attacks are conducted against the RC4 stream cipher itself, but while SWEET32 attacks

target a 64-bit block cipher, the use of CBC cipher mode is required, so we would hesitate to call this a direct attack against a cryptographic algorithm. When the SWEET32 attack appeared, an Internet draft (draft-kaduk-kitten-des-des-des-die-die-die) suggesting that the use of TripleDES be terminated was reconsidered. Similar to RFC7465 that removed RC4 as a usable algorithm, discussions regarding TripleDES took place in the CFRG, but did not make it to RFC status.

II. Changes in Browser Vendor Support Status

From the perspective of rating websites, it is now possible to know the status of a server easily through a web browser. One way this is done is through security indicators in the area where the URL is displayed. For example, a green bar is shown when a user accesses a server that implements EV SSL certificates. In Chrome, the method for displaying these security indicators changed in version 52 (for Macintosh desktops only; version 53 in other environments). The changes were made after a presentation at an international conference on usability security held in June 2016 [2], resulting in an improved interface being introduced. This paper took the approach of providing test subjects with several variations of the security indicator icons that indicate the status of a SSL/ TLS connection, and had them choose what they felt was the best, and the test results were implemented into Chrome. The icons shown have meanings relating to the trustworthiness of connections and servers, and different icons are used accordingly. There are separate icons for a proper HTTPS communication with a valid EV SSL certificate, but also HTTPS communications with minor errors, and HTTPS communications with major errors. Of these, HTTPS communications with minor errors is shown as mixed content (HTTP content mixed with HTTPS content) where there is content that HTTP points to within the HTML content, one of the typical example is web page embedded a ”search form” that jumps to HTTP site.

Browser vendors have appealed to sites that are causing mixed content to rectify the matter[3][4]. Prior to the aforementioned update, the icon provided a neutral impression, but the paper resulted in the selection of an icon that did not appear to be critically important but provided a negative impression to get the attention of users. Because Web administrators have not been able to catch up with these browser vendor changes, a number of SSL/TLS servers are unintentionally sending out HTTPS content that results in mixed content errors, so network operators and also contents holders should check the status of Web pages via Major browsers.

III. SSL/TLS versions status

This section provides results of a transitioning to new (and also secure) versions of TLS on (α) Alexa Top 20,000 sites, (β) .jp domain sites in Alexa Top 1M list and (γ) websites of local banks in Japan.

version	2014-04	2014-11	2015-01	2015-06	2017-04
SSL2.0	05.23	01.73	01.62	01.23	00.4
SSL3.0	98.57	37.42	33.78	23.67	09.3
TLS1.0	99.48	99.69	99.75	99.39	97.1
TLS1.1	56.66	72.66	74.46	80.83	90.8
TLS1.2	60.66	76.42	78.37	83.98	93.4

TABLE I. SSL/TLS versions status - (α) Alexa top sites

For servers belonging to the association of the banks, login sites for handling more important information are served by FQDN different from Top FQDN (refers to an FQDN of the site announced widely on paper medium and the like). In the top FQDN servers, there was a tendency similar to that of the .jp domain, but as shown below it turned out that the server operation was based on a secure setting. Approximately half of them are using outsourced servers, and the number of servers to be investigated has drastically decreased.

version	(α) Alexa	(β) .jp domain	(γ) Top FQDNs	(γ) Login sites
SSL2.0	00.4	04.2	04.3	00.0
SSL3.0	09.3	30.6	34.8	05.2
TLS1.0	97.1	99.2	100.0	100.0
TLS1.1	90.8	62.8	67.0	43.1
TLS1.2	93.4	65.9	69.6	62.1

TABLE II. Versions status at April 13th, 2017

At the login sites, there were no servers causing mixed content error. We also verified all the paths to access the Top FQDN via HTTP and follow the login site, but it was almost correctly designed, except for one case of entering login information from Non-SSL (HTTP) site. Most of the login sites use EV SSL certificates, but cases where outsourced SIer is displayed on security indicator of the browser are scattered a lot. On the other hand, outsourcing companies that correctly understand these ”search form” issues have deployed the original bank certificates. It is considered a good example of correctly capturing user needs.

IV. Conclusion: Ideal HTTP / HTTPS site design

Based on the above, we suggest the ideal server design as follows:

- When accessed by HTTP with Top FQDN, when forwarding to HTTPS, correct certificate should be returned so as not to cause browser error.
- The HTTPS site of the Top FQDN should separate the HTTP site from the contents.
- When accessed by HTTPS with Top FQDN, in the case of redirecting to HTTP, the certificate of Top FQDN should be placed under browser's certificate store. (make sure no error message occurs)
- The link to the login site should be done from the HTTPS page.
- The EV SSL certificate of the login page is not the name of the contractor of the outsourcing destination, but the official name of the site should be written.

References

[1] CELLOS consortium, Publication https://www.cellos-consortium.org/index.php?Publication

[2] Adrienne Porter Felt et al., ”Rethinking Connection Security Indicators”, SOUPS2016, https://www.usenix.org/system/files/conference/soups2016/soups2016-paper-porter-felt.pdf

[3] Google Developers - Web Fundamentals, Preventing Mixed Content https://developers.google.com/web/fundamentals/security/prevent-mixed-content/fixing-mixed-content

[4] Mozilla support, Mixed content blocking in Firefox https://support.mozilla.org/en-US/kb/mixed-content-blocking-firefox