Longitudinal Study of the Use of Client-side Security Mechanisms on the European Web

Ping Chen†, Lieven Desmet†, Christophe Huygens†, Wouter Joosen†

†iMinds-DistriNet, KU Leuven, 3001 Leuven, Belgium
†{firstname.lastname}@cs.kuleuven.be

ABSTRACT

As the web rapidly expands and gets integrated into all kinds of business, browsing the web has become an important part of people's daily lives. With the rising importance of various web applications sit in a browser, attackers also shifted their focus towards client-side attacks. To defend against these attacks, numerous client-side security mechanisms for the browser are proposed. The presence of these mechanisms on a website can be used as an indicator of the security awareness and practices of that website.

In this paper, through a large-scale analysis of more than 18,000 European websites over two years, we analyze the longitudinal trends of the adoption of client-side security mechanisms. We validate that the most popular websites were adopting new security features quicker that less popular websites in the two year timeframe. By examining the websites based on their business vertical, we observe that the websites in the *Finance* and *Education* category are outperforming other verticals in the data set, with respect to the usage of client-side security mechanisms.

Keywords

client-side security, empirical research, European web

1. INTRODUCTION

The web is constantly evolving, with new technologies such as HTML5 and CSS3 getting widely used and supported, which provide internet users richer experience. In the mean time, the attacks on the web are also changing, shifting from server exploitation such as SQL injection to client-side attacks such as XSS and Man-in-the-Middle (MitM) attacks such as SSL-stripping. In respond to this trend, various client-side security mechanisms are developed, such as Content Security Policy (CSP) and HTTP Strict-Transport-Security (HSTS). These client-side security mechanisms are sever-driven, but requires the browser to enforce them. The presence of these mechanisms on a website can be used as an indicator of the security awareness and practices of that website.

This paper tries to give an overview of the adoption of client-side security mechanisms on the web, and provide a state-of-practice reference model of web security for website operators. To achieve this, we crawled more than 18,000 European websites in a two-year period. With the gathered data, we analyze the evolution of the usage of client-side security mechanisms, and use a security scoring system to compare a website to its peers (based on business vertical or popularity), in order to provide a web security baseline for website operators.

Our main contributions are the following: (1) We report the usage of seven client-side security mechanisms on European web in September 2013 and September 2015, and analyze the evolution of adoption (i.e., identify which security features are being adopted over time); (2) We propose a web security scoring system to compare the security posture (i.e., the practice of the usage of client-side security features) among different websites, and among countries, business sectors; (3) We provide a web security baseline and maturity model for website operators, by applying the web security scoring system to a set of websites.

2. DATA COLLECTION

To study the security posture of the European web, the popular websites from the 28 member states in the EU are chosen to represent the European web. For each EU country, we selected the top 1,000 websites ending with the corresponding ccTLD (country code top-level domain) from Alexa's list of the top 1 million sites [1][1]. As a result, we have a set of 23,050 European websites.

We then obtained up to 200 webpage URLs for each website by querying the Bing search engine [2] for the popular webpages of each website. After the webpage URLs are obtained, PhantomJS, a headless browser, is used to visit the URLs and retrieve data from webpages. By loading every webpage within PhantomJS [4], we mimicked the behavior of a regular visitor using a Chrome browser.

After we have crawled all the URLs, we remove the websites with less than 50 successfully crawled pages from our dataset. Then we have a dataset of 20,157 websites in 2013, and a dataset of 18,074 websites in 2015, as shown in Table 1. The 2015 dataset is smaller than 2013 dataset, which is due to that some domains are disappeared or redirected to other domains.

[1]Alexa top 1 million list in September 2013

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
DOI: http://dx.doi.org/10.1145/2872518.2888605 .

Time	# of sites	# of pages	avg. # of pages/site
Sept. 2013	20,147	3,499,080	174
Sept. 2015	18,074	2,992,395	166

Table 1: Overview of datasets

3. SECURITY FEATURES AND SCORING SYSTEM

3.1 Security features

In this study, we focus on seven security features that can all be passively detected. We grouped them into three categories:

- 3 features that contribute to **Secure Communication**
- 3 features that mitigate **Cross-Site Scripting**
- 1 feature that enables **Secure Framing**

3.1.1 Category 1: Secure Communication

The Secure Communication category groups three security features: HTTPS support, Secure Cookies, and HTTP Strict Transport Security (HSTS).

HTTPS Support. The HTTPS protocol is the standard solution for securing web traffic, which guarantees the confidentiality and integrity of web communications by adding the security capabilities of SSL/TLS to HTTP. It also provides website authenticity with the CA/B (Certificate Authority/Browser) trust model.

Secure Cookies. HTTPS websites should set the `Secure` attribute when sending cookies to a user's browser, which can prevent cookies from being intercepted by an active network attacker. Although the traffic between a web server and a browser is encrypted when using HTTPS, the cookies stored in the browser are not, by default, limited to an HTTPS context. Thus an active network attacker can intercept any outbound HTTP request from the browser and redirect that request to the same website over HTTP in order to reveal the cookies [5]. By setting the `Secure` attribute, the scope of a cookie is limited to secure channels, thus stopping browsers from sending cookies over unencrypted HTTP requests.

HTTP Strict-Transport-Security (HSTS). HSTS [6] is designed to mainly prevent SSL-stripping attacks where a secure HTTPS connection is downgraded to a plain HTTP connection by the attacker. Set by a website via a HTTP response header field (`Strict-Transport-Security`), HSTS specifies a period of time during which the user's browser is instructed that all requests to that website need to be sent over HTTPS, regardless of what a user requests. The HSTS Policy helps protecting website users against both passive eavesdropping, as well as active Man-in-the-Middle (MITM) attacks [7].

3.1.2 Category 2: XSS Mitigation

The XSS Mitigation category groups three security features: HTTPOnly Cookies, X-Content-Type-Options (XCTO), and Content Security Policy (CSP).

HttpOnly Cookies. First introduced in Internet Explorer (IE) 6 SP1, the `HttpOnly` attribute is designed to mitigate the risk of malicious client-side scripts accessing sensitive cookie values. Cookies are accessible to JavaScript code by default, which allows attackers to steal the cookies via an XSS attack. Using the `HttpOnly` attribute in a `Set-Cookie` header restrict the access of that cookie to the HTTP(S) protocol, making it inaccessible to client-side JavaScript [5].

X-Content-Type-Options (XCTO). Internet Explorer has a MIME-sniffing feature that will attempt to determine the content type for each downloaded resource. This feature, however, can lead to security problems for servers hosting untrusted content. To prevent Internet Explorer from MIME-sniffing, thus reducing exposure to attacks, a web server can send the `X-Content-Type-Options` response header with the `nosniff` value.

Content Security Policy (CSP). CSP provides a standard HTTP response header (`Content-Security-Policy`) that allows a webpage to declare approved sources of content that browsers should be allowed to load on that specific page. Whenever a requested resource originates from a source that is not defined in the CSP, it will simply not be loaded [10]. For example, if the policy does not allow in-line JavaScript, then, even if an attacker is able to inject malicious JavaScript in the webpage, the injected code will not be executed.

3.1.3 Category 3: Secure Framing

The Secure Framing category reports on the use of X-Frame-Options (XFO).

X-Frame-Options (XFO). The HTTP response header `X-Frame-Options` is designed to mitigate Clickjacking attacks [8]. In a Clickjacking attack, the attacker redresses the user interface of website A with transparent layers, and then trick the user into clicking on a button on an embed page from website B when they were intending to click on the the same place of the overlaying page from website A. To stop Clickjacking attacks, the `X-Frame-Options` header can be used to instruct a user's browser whether a certain page is allowed to be embedded in a frame. For example, if the `X-Frame-Options` header's value is `DENY`, then the browser will prevent the page from rendering when embedded within a frame.

3.2 Web Security Scoring System

In order to compare the security level among different websites, and among countries, business verticals (represented by the websites in each country or business vertical), we developed a web security scoring system that gives a quantitative security score for each website.

For each (group of) website(s), we define the overall security score (*OverallScore*) as a weighted average of three distinct subscores:

$$\begin{aligned}\textbf{OverallScore} = &\frac{40}{100} \times SecureCommunicationScore \\ &+ \frac{40}{100} \times XSSMitigationScore \\ &+ \frac{20}{100} \times SecureFramingScore\end{aligned}$$

As part of of scoring system, we assess for each security feature how well the (group of) website(s) is doing compared to websites in the full dataset. For instance, we want to grade a website with a score 0.61 to the feature HTTPS, if the website outperforms 61% of the websites in our dataset (i.e. by having a higher percentage of pages over HTTPS). The scores of the individual features are then combined to provide a metric for the three subscores.

More concretely, we apply the following approach:

1. For each security feature, we compute an empirical cumulative function (ECDF) for all websites. The ECDF is computed based on the percentage of webpages having that feature on a particular website.

2. This computed ECDF is used to calculate an ECDF value per website and per feature.

3. The subscores are calculated by applying a weighted averages of the ECDF values.

The weight given for each feature reflect the relative importance and maturity of the feature in each category. The more fundamental and matured feature get a relatively higher weights. In particular, the following weights are used to calculated the three subscores:

Secure Communication Score. This subscore is measured by applying a weighted average of the HTTPS, HSTS, and Secure Cookies usage.

$$SecureCommunicationScore = \frac{45}{100} \times HTTPS + \frac{25}{100} \times SecureCookies + \frac{30}{100} \times HSTS$$

XSS Mitigation Score. This subscore measured by applying a weighted average of the HttpOnly Cookies, XCTO, and CSP usage.

$$XSSMitigationScore = \frac{50}{100} \times HttpOnlyCookies + \frac{25}{100} \times XCTO + \frac{25}{100} \times CSP$$

Secure Framing Score. This subscore is measured by the XFO usage.

$$SecureFraming = \frac{100}{100} \times XFO$$

4. GENERAL FINDINGS

4.1 Overview on the use of security features on European web

Table 2 gives an overview on the use of security features on European web in 2013 and 2015. It clearly illustrates that the web security on the European Web did improve, as each of the security features have been adopted in 2015 by a larger fraction of websites than in 2013.

Security feature	% of websites		Improvement
	Sept. 2013	Sept. 2015	
HTTPS Support	22.96%	32.40%	9.44%
Secure Cookies	5.86%	7.56%	1.70%
HSTS	0.49%	4.30%	3.81%
HttpOnly Cookies	36.52%	43.86%	7.34%
XCTO	2.24%	6.82%	4.58%
CSP	0.05%	0.43%	0.38%
XFO	4.80%	14.93%	10.13%

Table 2: Overview of the use of security features on European web

The above table gives an overview for the whole European Web. We also assess to what extent the security of a particular website did improve over time, by measuring for each website if it adopts more security features over time or not. Since none of the websites have all seven security features enabled in 2013, there is space for improvement for all the websites.

By doing this, we found 6,512 (36%) websites that adopted more security features in 2015 than what they already have in 2013. And there are 22 websites that have all seven security features adopted in 2015.

4.2 Websites that adopted more security features

In this section, we investigate the relationship between the adoption of security features on a website and its popularity (measured by its Alexa global rank [1]), its sector (derived from McAfee's TrustedSource Web Database [3]). We expect that higher ranked popular website and websites belonging to critical sectors such as finance and online shopping, might have more incentive to adopt security features in order to protect their asset, compared to the less-known or less-valuable websites.

To confirm this hypothesis, we first use Point-biserial correlation to study the correlation between the adoption of security features in a website and its Alexa rank. Generally, Pearson product-moment correlation coefficient (Pearson's r) is widely used in statistics as a measure of the degree of linear dependence between two quantitative variables. In our case, the adoption of security feature is a binary choice, thus we use the Point-biserial correlation coefficient. The Point-biserial correlation coefficient is a special case of Pearson in which one variable is quantitative and the other variable is dichotomous. The result of Point-biserial correlation varies between -1 and $+1$, and a positive coefficient implies that as one variable increases, the other variable also increases and vice versa. When using Point-biserial correlation to test statistical dependence, we set the significance level to 5%. The p-value is calculated using Student's t-distribution. We accept the hypothesis only if the p-value is smaller than the significance level.

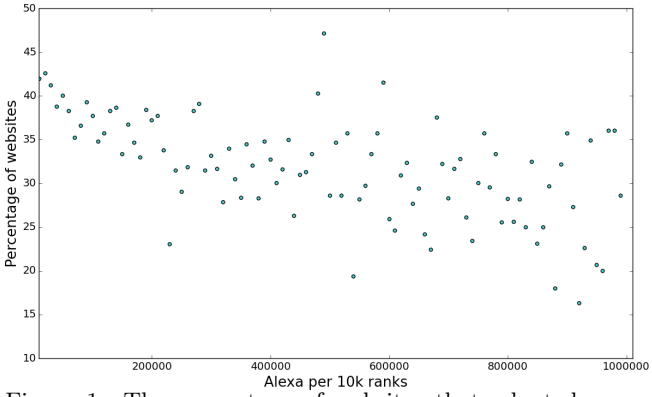

Figure 1: The percentage of websites that adopted more security features in 2015 versus 2013, plotted per 10k Alexa ranks

The resulting Point-biserial correlation coefficient is -0.08, with p-value 6.24×10^{-29}, which indicates a negative correlation, hence it confirms our hypothesis that higher ranked websites tend to adopt more security features. To better illustrate this correlation, for per 10,000 Alexa ranks, we calculate the percentage of websites that belongs to that rank range, which have adopted more security features, as shown in Figure 1. We can observe a downtrend for the percentage of websites that adopted more security features over the Alexa ranks.

As for the relationship between the adoption of security features in a website and its sector, we calculate the percentage of websites that adopted more security features in each sector. Figure 2 shows the top 10 sectors that have larger percentage of websites adopted more security features over time. It comes as no surprise that the *Finance* vertical is the best performing category. And among the 22 websites that have all seven security features enabled in 2015, half of them (11 websites) are from the *Finance* sector.

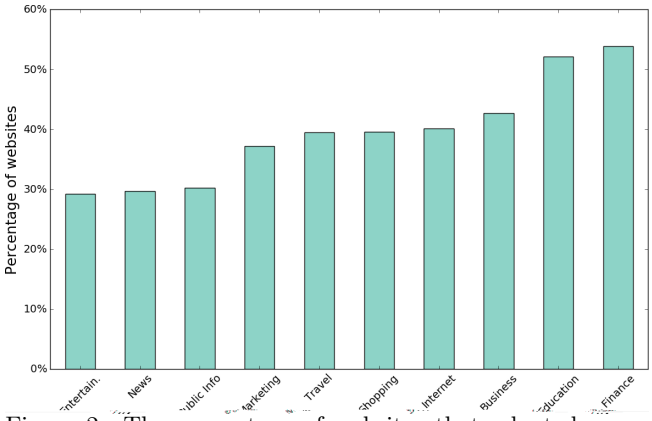

Figure 2: The percentage of websites that adopted more security features in 2015 versus 2013, grouped per business vertical

5. WEB SECURITY SCORE ANALYSIS

In the previous section, we assess to what extent the security of a particular website did improve over time, by measuring for each website if it adopts more security features over time or not. In this section, we investigate how consistently a security features is applied on a given website, by calculating ECDF-based security scores (as explained in Section 3.2), which essentially compare the usage of a security feature on a particular website with the usage on other websites in the dataset.

Correlation 2013 Dataset		Correlation 2015 Dataset	
coefficient	p-value	coefficient	p-value
-0.11	1.81×10^{-59}	-0.08	3.84×10^{-31}

Table 3: Correlation between the *OverallScore* in a website and its Alexa rank

5.1 EU Web Security Score, in terms of website popularity

To assess the web security score in terms of website popularity, the websites are grouped per 10,000 Alexa ranks, and the average score is calculated for websites that belongs to that rank range. Figure 3 shows the average *OverallScore* for per 10k Alexa ranks.

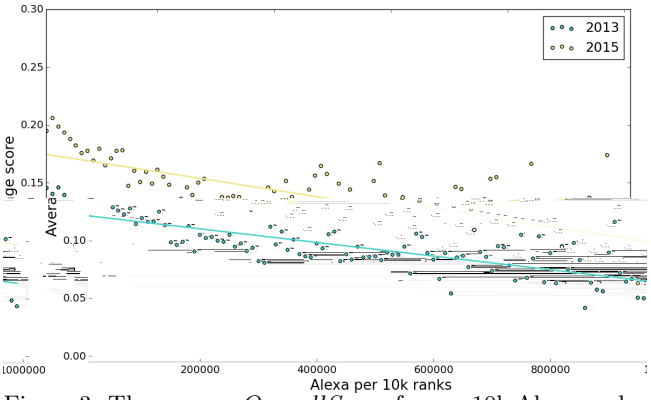

Figure 3: The average *OverallScore* for per 10k Alexa ranks

Figure 3 hints that higher ranked websites tend to have higher score. To confirm this assumption, the Spearman correlation is used to assert the correlation between the *OverallScore* in a website and its Alexa rank for the 2013 dataset and 2015 dataset (as listed in Table 3).

Spearman's rank correlation coefficient is a nonparametric measure of the monotonicity of the relationship between two variables. It is defined as the Pearson correlation coefficient between the ranked variables. However, unlike the Pearson correlation, the Spearman correlation does not assume that both variables are normally distributed. It is a nonparametric statistic, which do not rely on assumptions that the dataset is drawn from a given probability distribution. The result of Spearman correlation varies between -1 and $+1$, and a positive coefficient implies that as one variable increases, the other variable also increases and vice versa. When using Spearman correlation to test statistical dependence, we set the significance level to 5%. The p-value is calculated using Student's t-distribution. We accept the hypothesis only if the p-value is smaller than the significance level.

As expected from Figure 3, there is negative correlation between the *OverallScore* in a website and its Alexa rank (see Table 3), and this correlation also holds for all three sub scores. This correlation is consistent with the correlation that higher ranked websites tend to adopt more security features, as we found in the previous section.

5.2 Web Security Score per business vertical in EU

In this section, we compare the security evolution of the websites per business vertical. For the ten most popular business vertical, the average score is calculated for websites that belongs to that business vertical. Figure 4 shows the average *OverallScore* for 10 business verticals, sorted by their 2013 security score, to easily identify business verticals that got better than their adjacent peers.

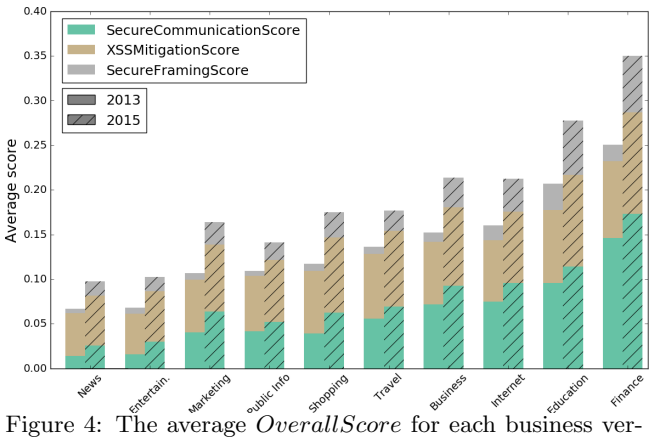

Figure 4: The average *OverallScore* for each business vertical

The *Education* and *Finance* verticals are the two best performing categories in each of the subscores and the Overall score. In addition, we can observe that the *Business*, *Shopping*, and *Marketing* sectors improved a lot and eventually caught up their neighbors.

5.3 Web Security Score per country in EU

In this section, we compare the security evolution of the websites per country. For each EU country, the average score is calculated for websites that belongs to that country. Figure 5 shows the average *OverallScore* for 25 EU countries. Cyprus(.cy), Malta(.mt) and Luxemburg(.lu) were removed from the dataset, as the number of websites in these countries were less than 100. The countries in Figure 5 are sorted by their 2013 security score, to easily identify countries that got better than their adjacent peers.

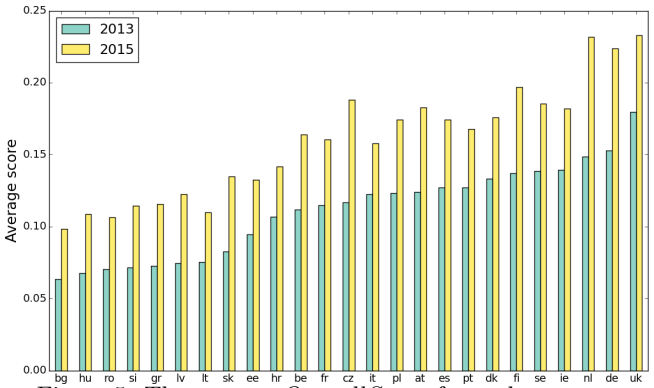

Figure 5: The average *OverallScore* for each country

6. HTTPS MIGRATION ANALYSIS

HTTPS in the standard solution for securing web traffic nowadays, though it increases performance overhead and operating costs. Among the 6,512 websites that adopted more security features, 2,383 of them adopted HTTPS. We call these websites the newly adopted HTTPS sites.

Figure 6 shows the top 10 sectors that have the most percentage of websites with HTTPS support. We can see that *Finance* and *Gambling* are the best two verticals in 2013, but *Gambling* was caught up by *Gov./Mil.* and *Education* in 2015.

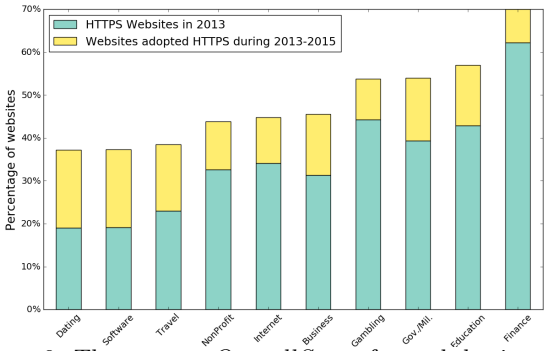

Figure 6: The average *OverallScore* for each business vertical

While HTTPS already provides securing communication, it would be better to also implements `HSTS` and `Secure Cookies` to have stronger protection against MITM attacks. In this section, we also investigate whether the newly adopted HTTPS sites also implement `HSTS` and `Secure Cookies`, when migrating to HTTPS.

As shown in Table 4, only 7.7% of newly adopted HTTPS sites have HSTS implemented, which is less than the overall percentage (11.5% of all HTTPS sites in 2015). And the use of `Secure Cookies` in newly adopted HTTPS sites is also less than the overall percentage. This indicates that the newly adopted HTTPS sites are not more security conscious than the websites having HTTPS already for a longer period. In other words, the adoption of HTTPS does not necessary lead to the adoption of `Secure Cookies` and `HSTS`.

	HTTPS sites in 2015	Newly adopted HTTPS sites
HSTS	11.5%	7.7%
Secure Cookies	22.8%	11.2%

Table 4: HTTPS

7. RELATED WORK

Client-side web security is becoming increasing important nowadays. In [9], De Ryck et al. discussed various client-side vulnerabilities and attacks, and enumerated best practices for web security with existing countermeasures and emerging mitigation techniques. Weissbacher et al. [12] addressed client-side validation (CSV) vulnerability and proposed a system (ZigZag) to strengthen JavaScript-based web applications against client-side validation attacks.

This paper is built upon one of our previous works by Van Goethem et al [11], in which the authors conducted a security assessment for more than 22,000 European websites

in 2013, showing that such a large-scale security analysis of the web is achievable, albeit challenging.

8. CONCLUSION

In this paper, we have compared the security state of practice in the European Web, both in September 2013 and September 2015. To do so, we crawled about 3 million web pages of 18,000 websites. Based on analysis of available data of the crawling experiment, we could observe the following longitudinal trends:

First, we have observed that the most popular websites (according to the Alexa ranking) have a higher web security metric that less popular websites. Moreover, we could validate that the most popular websites were adopting new security features quicker that less popular websites in the two year timeframe.

Second, by examining the websites based on their business vertical, we can state that the websites in the Finance and Education category are outperforming other verticals in the data set, with respect to the web security metric.

We also proposed a web security scoring system to compare different websites. The scoring system can be used to establish a web security baseline among a set of websites, and this might help website operators to consider the adoption of security features.

Acknowledgements

We would like to thank the reviewers for their comments. This research is partially funded by the Research Fund KU Leuven, iMinds, IWT, and by the EU FP7 projects WebSand, NESSoS and STREWS. With the financial support from the Prevention of and Fight against Crime Programme of the European Union (B-CCENTRE).

References

[1] Alexa top 1 million sites. `http://s3.amazonaws.com/alexa-static/top-1m.csv.zip`.

[2] Bing Search API. `http://datamarket.azure.com/dataset/bing/search`.

[3] Mcafee trustedsource web database. `https://www.trustedsource.org/en/feedback/url`.

[4] Phantomjs: Headless webkit with javascript api. `https://www.phantomjs.org/`.

[5] A. Barth. HTTP state management mechanism. *IETF RFC*, 2011.

[6] J. Hodges, C. Jackson, and A. Barth. HTTP strict transport security (HSTS). *IETF RFC*, 2012.

[7] Moxie Marlinspike. New tricks for defeating ssl in practice. *Blackhat*, 2009.

[8] D. Ross and T. Gondrom. HTTP Header X-Frame-Options. *IETF RFC*, 2013.

[9] Philippe De Ryck, Lieven Desmet, Frank Piessens, and Martin Johns. *Primer on Client-Side Web Security*. Springer, 2014.

[10] Sid Stamm, Brandon Sterne, and Gervase Markham. Reining in the web with content security policy. In *Proceedings of the 19th international conference on World wide web*, pages 921–930. ACM, 2010.

[11] Tom van Goethem, Ping Chen, Nick Nikiforakis, Lieven Desmet, and Wouter Joosen. Large-scale Security Analysis of the Web: Challenges and Findings. In *Proceedings of the 7th International Conference on Trust and Trustworthy Computing*, 2014.

[12] Michael Weissbacher, William Robertson, Engin Kirda, Christopher Kruegel, and Giovanni Vigna. Zigzag: Automatically hardening web applications against client-side validation vulnerabilities. In *24th USENIX Security Symposium*, pages 737–752, 2015.