Web Encryption Analysis of Internet Banking Websites in Thailand

Suphannee Sivakorn, Patsita Sirawongphatsara and Nuttaya Rujiratanapat
Department of Computer Science, Faculty of Science and Technology
Rajamangala University of Technology Tawan-ok, Chon Buri, Thailand
Email: {suphannee_si, patsita_si, nuttaya_ru}@rmutto.ac.th

***Abstract*—With Thailand rapidly moving to a full internet banking ecosystem, the demand for online security has never been needed more than it is today. As the security and privacy of internet users depend on HTTPS, a web encryption protocol, for securing communication between users and web servers, HTTPS is essentially the center of the web ecosystem today. Unfortunately, despite the increasing number of HTTPS adoptions, numerous studies have shown that a large number of websites have adopted HTTPS incorrectly, rendering users vulnerable to information leakages e.g., eavesdropping and man-in-the-middle attacks. The correctness of HTTPS deployment is even far greater for internet banking services due to carrying user's sensitive information and being prime targets for criminal activities.**

In this paper, we present WEAPONS, a novel black-box testing tool for evaluating the completeness and correctness of web encryption deployment including the deployment of HTTPS, and web encryption-related mechanisms i.e., HSTS, secure cookie, HTTPS redirect, HSTS preload. We use WEAPONS to conduct an assessment of 9 popular internet banking websites in Thailand during January - February 2020. We demonstrate that WEAPONS is able to find HTTPS deployment incorrectness. Several of these weaknesses can expose the affected services to man-in-the-middle attacks and sensitive data exposure.

I. Introduction

Enabling users to securely communicate sensitive data online was a crucial foundation of the web ecosystem today. HTTPS was originally introduced in 1994 by Netscape Communications [1]. This marked the birth of encryption on web protocol allowing the basis of securing many aspects of the internet today e.g., authentication, secure data transfer, secure download, private browsing. With an increasing part of our daily life revolving around the Internet and a large amount of personal data being uploaded to websites, ensuring the user privacy of digital communications has become a pressing matter. Several of recent studies have pointed out the importance of securing web connection from adversaries by demonstrating how easily for attackers to hijack a user's session and information [2]–[9]. Nevertheless, many major services carelessly continue to serve content over unencrypted connections – not enforcing encrypted connections as demonstrated in a number of recent studies [5], [6], [9], [10]. This is due to ranging from changing infrastructure, increasing costs, maintaining support for legacy clients [11] to simply misconfigurations of related security mechanisms [9], [12].

Things become worse when it comes to any security vulnerabilities on banking applications, as financial institutions have always been at the center of malicious and fraudulent activity [13]–[16]. Without a doubt, banking institutions must ensure the correctness of their security technology and systems to continuously ensure and secure customers' trust. With Thailand moving to a full internet-based financial ecosystem, the demand for online security and privacy has never been needed more than it is today.

In this paper, we present **WEAPONS**, a **W**eb **E**ncryption **A**nalysis **P**rogram for **ON**line **S**ervices. WEAPONS is a novel black-box testing framework for evaluating the completeness and correctness of web encryption deployment including the HTTPS, web encryption-related mechanisms i.e., HTTPS redirection, HSTS, secure cookie, HSTS, and HSTS preload. To evaluate the performance of WEAPONS, we apply WEAPONS to analyze and assess the state of web encryption of 9 popular internet banking websites in Thailand during January - February 2020. To this end, we found a number of misconfigurations that potentially lead to sensitive data exposure scenarios.

Overall, our goal is twofold. First, to understand the state of web encryption of the Thai internet banking website. Second, to create automated testing for evaluating the website encryption for identifying misconfigurations, which could lead to sensitive data exposure.

The main contributions of this paper are as follows.

- To the best of our knowledge, our study is the first in-depth study of internet banking websites in Thailand in terms of web encryption security.
- We design and implement WEAPONS, an automated tool for evaluating the completeness of web encryption. The tool can be used for efficient testing and auditing websites by identifying misconfigurations of major web encryption-related mechanisms, namely, HTTPS, HSTS, secure cookies.
- We evaluate WEAPONS on 9 popular online-banking services. WEAPONS is able to discover flaws in deployment which are potentially lead to session hijacking attacks and information leakages.

978-1-7281-0719-6/19/$31.00

II. Background

In this section, we first provide a summary of the web encryption, web encryption security mechanisms, and an overview of sensitive data exposure.

A. Web Encryption

To ensure user's security and privacy on the web, there are a number of standard protocols and mechanisms in both web client (browser) and server side.

1) HTTPS: The Hypertext Transfer Protocol Secure is a security protocol designed as an extension of HTTP for secure communication. HTTPS is an implementation of SSL/TLS over HTTP [17]. HTTPS enables data to be sent using encrypted protocol (HTTPS), this, in turn, renders three layers of protections i.e., data confidentiality, integrity and authentication. Sivakorn et al. [9] pointed out that HTTPS often interpret as add-on security and is not deploy ubiquitously. However, it is recommended websites to deploy HTTPS on all pages regardless of the content and sensitivity of it [18].

2) HTTPS Redirect: By default, typing in domain name without https:// into the browser directs users to HTTP. For example, www.example.com sends HTTP request http://www.example.com [9]. Therefore, it is up to websites to redirect browsers to HTTPS as soon as possible. Usually, this is being done by HTTP redirects e.g., 301 Permanent, 302, 307 Temporary. However, it is recommended to deploy 301 Permanent redirect, since it allows browsers to permanently honor HTTPS redirect and allows search engines to link to the HTTPS version of the website [18].

3) HSTS: The HTTP Strict Transport Security [19] is an HTTP response header that allows websites to inform web browsers to only connect over HTTPS without relying on HTTPS redirection from the server-side. When the HSTS website is accessed over HTTP, the browser knows that the site has HTTPS and internally redirects to HTTPS honoring the Strict-Transport-Security header, this means no HTTP is sent out.

Without HSTS, visitors may communicate with HTTP (non-encrypted) version of the website e.g., before HTTPS redirection, hidden HTTP requests such as HTTP requests from mixed content [20]. This creates an opportunity for man-in-the-middle attacks e.g., SSL Strip [3] as well as other session hijacking attacks. We detail related attacks in Section II-B.

A number of studies have demonstrated scenarios where an attacker could bypass HSTS mechanism [21], [22]. Kranch and Bonneau [12] studied the adoption of HSTS and certificate pinning on the top one million websites. Based on their findings, they reported a lack of understanding by web developers on the proper use of these mechanisms, as they often use them in illogical or incorrect ways.

HSTS is enabled through configuring HTTP response header. The header must be set by HTTPS response as it will be ignored by the browser when the site is accessed using HTTP. This is because an attacker may intercept HTTP connections and inject the header or remove it. For better understanding of our evaluation presented in Section IV, here we describe the three HSTS directives and their recommended configurations.

- *max-age:* The time in seconds that the browser will enforce the site to be only connected with HTTPS. This must be any positive integer value. This duration will be updated when a new Strict-Transport-Security header sent to the browser. There is no official recommendation for the duration of `max-age`. Nevertheless, the longer `max-age` duration will reduce the number of HTTP connections. The most common default setup is at the duration of one year (`max-age: 31536000`).
- *includeSubDomains:* The `includeSubdomains` is an optional header directive. HSTS is set based on the subdomain and domain found in the website URL. For example, when a user accesses the website https://www.example.com/foo with HSTS enabled, the HSTS is enabled on any pages accessed via www.example.com. HSTS, however, is not enabled on pages on upper-level domain (example.com) and other pages on the subdomains e.g., bar.www.example.com. With the `includeSubdomains` directive, the browser enables HSTS on all subdomains – any pages on *.www.example.com. The `includeSubdomains` is recommended as it will enforce HTTPS any pages on subdomains. When possible, `includeSubdomains` should be set in the domain-name level (e.g., example.com) to protect all pages on the domain name.
- *preload:* The `preload` is another optional header directive. This directive currently maintained by Google [23]. With the `preload` directive, the HSTS is pre-loaded in the browser. Therefore, the HSTS preload websites will never be connected with HTTP. In order to deploy this directive correctly, the website needs to be in the preloaded list by satisfying all submission requirements including having a valid certificate, HTTPS redirection, all subdomains over HTTPS, and HSTS header with `max-age` at least one year, `includeSubdomains` and `preload` directives [23].

4) Secure Cookie: A HTTP cookie is a piece of small data that a server sends to the user's web browser. When receiving an HTTP request, a server can send `Set-Cookie` header with HTTP response [24]. The cookie is stored by browser by default and sent with HTTP request made to the same server in the `Cookie` header. HTTP cookies are widely use for authentication purposes. This is due to the fact that it can be used as a session ID after authentication process. Therefore, HTTP cookies must be stored and transmitted securely to help mitigate session hijacking attacks. As part of the RFC 6265, the secure flag is one of an option provided. Cookies with secure flag is sent to the web server only over HTTPS [24]. Thus, websites with sensitive contents and authentications are strongly recommended to employ `Secure` flag.

B. Sensitive Data Exposure

1) Session Hijacking Attacks: The security problem we study in this work is mainly the result of websites not

enforcing encryption across all pages and subdomains, so as to adequately protect cookies. Marlinspike introduced the SSL Striping attack [3]. The author described the danger of downgrading attack by hijacking the user's first request to the web server before switching to HTTPS. HSTS was later purposed to mitigate such attacks. Previous work has shown the risks of supporting mixed-content websites, where pages accessed over HTTPS also include HTTP content which is fetched over HTTP [20]. Englehardt et al. [5] discussed how third-party HTTP cookie leaked over HTTP connections can be used for web tracking. Their results highlight the threat of unencrypted connections. Liu et al. [10] studied PII being transmitted in unencrypted network traffic.

2) HTTP Cookie-related Issues: Zheng et al. [7] presented a study on the HTTP cookie injection attacks. While cookies have the `Secure` flag that can prevent browsers from sending them over unencrypted connections, there is no provision to ensure that such cookies are also set only over HTTPS connections. As a result, during an HTTP connection to a domain, an attacker on the network can inject cookies that will be appended in future HTTPS connections to that specific domain. Sivakorn et al. [25], demonstrated how HTTP cookies could be used for influencing Google's advanced risk analysis system and bypassing reCAPTCHA challenges. Castelluccia et al. [6] pointed out the problem of privacy leakage that can occur when personalized functionality is accessible to HTTP cookies. The authors demonstrated how adversaries could reconstruct a user's Google search history by exploiting the personalized suggestions of the search engine.

III. System Overview

This section describes the design and implementation of our testing platform, WEAPONS, Web Encryption Analysis Platform for Online Services.

Our system is implemented with Python 3.7 and mainly built on Selenium [26], an open-source browser automation framework and selenium-wire [27] which is an extension of Selenium for accessing HTTP requests. We opt for Chromium WebDriver, so we can leverage the rich functionality of the browser engine. Specifically, we build on top of Google Chrome Version 80.0. The WebDriver offers functionalities e.g., locating HTML DOM element, retrieving HTTP and HTTPS request headers and HTTP response headers. Therefore we are able to obtain a number of web encryption configurations i.e., HSTS, HTTP cookies, and HTTP redirection.

Figure 1 presents an overview of how WEAPONS is used to analyze and audit web encryption of a website. WEAPONS takes the website URL from the user as their input, then performs analysis on the website and later return the user with the results as soon as possible. Specifically, the system consists of five main steps. The first step is to obtain the website URL as the user input (1). Next, WEAPONS starts the Analyzer engine which is the main component of the system (2). The analyzer communicates with WebDriver as mentioned for navigating to the testing website and collecting

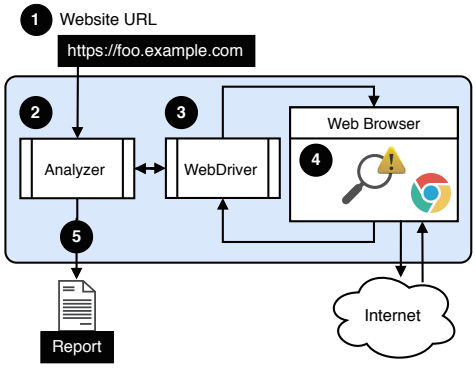

Fig. 1: Overview of WEAPONS Architecture

all necessary information (3). This is done by starting the web browser, perform automated tasks, and collect results according to commands given from the analyzer (4) in order to look for any potential vulnerabilities and misconfigurations in web encryption as described in Section II, which can be summarized as follows.

- The deployment of HTTPS
- HTTPS Redirection
- HSTS, HSTS directives, HSTS preload
- Secure flag of HTTP cookies

Finally, the system forwards all the findings and forward back to the analyzer engine for preparing human-readable output and generating an HTML report for further investigation (5). The whole process takes at most at about 60 seconds for each website. However, we also provide an option for users to reduce delays for waiting for the page to be fully loaded. To optimize time usages, WEAPONS can be run simultaneously with different WebDriver processes on the same experimental machine or server. Our findings from using WEAPONS to audit Thai internet-banking websites are presented in Section IV.

The project WEAPONS is open-source and can be accessed at https://github.com/ssivakorn/WEAPONS along with their source code, example reports, manual and example usages.

IV. Evaluation

To evaluate our testing framework, we select the 9 most popular internet banking services in Thailand. To obtain the most recent deployment status, this experiment was recently carried out during January - February 2020. Table I presents the list of selected services along with their internet banking service URL we audited.

A. Findings and Results

In this section, we detail our study on auditing internet banking services. We select these services from the most popular and well-known banks in Thailand. We discover some of HTTPS deployment misconfigurations which potentially cause information leakages. Table II presents an overview of the internet banking services and our results. We highlight our findings as follows.

TABLE I: internet banking Website and Their Service URL

Internet Banking Service	Bank	Service URL
Bualuang iBanking	Bangkok Bank	https://ibanking.bangkokbank.com
KTB Netbank	Krung Thai	https://www.ktbnetbank.com
TMB Direct	TMB	https://www.tmbdirect.com
Krungsri Online	Krungsri	https://www.krungsrionline.com
K-Cyber	Kasikorn	https://online.kasikornbankgroup.com
SCB Easy	Siam Commercial	https://www.scbeasy.com/v1.4/site/presignon/index.asp
KK e-Banking	Kiatnakin	https://ebanking.kiatnakin.co.th
GSB Internet	Government Saving Bank	https://ib.gsb.or.th
Thanachart iNet	Thanachart Bank	https://retailib.thanachartbank.co.th/retail/Login.do?action=form

TABLE II: Overview of the Audited Results of Internet-banking Websites in Thailand

Internet-banking Service	HTTPS	HTTPS Redirection	HSTS				Secure Cookies
			Header	IncludeSubdomains	Max-Age	Preload	
Bualuang iBanking	Yes	No	Yes	Yes	31536000	No	Mix
KTB Netbank	Yes	Yes	Yes	Yes	31536000	Yes	Full
TMB Direct	Yes	Yes	Yes	Yes	15683234	No	Mix
Krungsri Online	Yes	No	Yes	No	N/A	No	Mix
K-Cyber	Yes	No	Yes	Yes	31536000	No	Mix
SCB Easy	Yes	Yes	No	–	–	–	Mix
KK e-Banking	Yes	No	No	–	–	–	Mix
GSB Internet	Yes	Yes	Yes	No	15552000	No	Full
Thanachart iNet	Yes	No	Yes	Yes	63072000	Yes	Mix

Listing 1: KTB Netbank HSTS Record in the Preload List

```
{    "name": "ktbnetbank.com",
     "policy": "bulk-18-weeks",
     "mode": "force-https",
     "include_subdomains": true   }
```

1) Bualuang iBanking: Bualuang iBanking is an internet banking service from Bangkok Bank. The platform is designed to allow users to do banking transactions 24/7 and can be accessed with computers and tablets [28]. Our testing framework revealed that the Bualuang iBanking has supported HTTPS as well as HSTS with the `IncludeSubdomains` and `Max-Age` of a one-year duration. However, the service does not deploy HSTS preload.

2) KTB Netbank: KTB NetBank is an internet banking website from Krung Thai Bank. It is one of the online personal banking services provided by Krung Thai Bank [29]. We found that KTB Netbank is the most complete HTTPS deployment. This includes HTTPS, HTTPS redirect, HSTS, `includeSubdomains`, `Max-Age` of one-year duration and HSTS preload. The service is only one of the two internet banking websites we audited that have deployed HSTS preload. Listing 1 presents the HSTS record in the HSTS preload list.

From the HSTS record, the service has deployed the HSTS on their domain name (`ktbnetbank.com`) as well as all subdomains (`"include_subdomains": true`). This results in all of their subdomains including the login page (`www.ktbnetbank.com`) are always connected with HTTPS on supported browsers.

In addition to HTTPS and HSTS, all cookies we found when loading the login pages are set `Secure` flag. As mentioned, this ensures that any sensitive cookies will not be sent out through an HTTP connection.

Listing 2: Krungsri Online HTTPS Response Header.

```
HTTP/1.1 200 OK
content-type: text/html; charset=utf-8
transfer-encoding: chunked
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
cache-control: private
x-xss-protection: 1; mode=block
strict-transport-security:
x-ua-compatible: IE=edge
```

3) TMB Direct: TMB Bank has the TMB Direct as its internet banking service [30]. The website offers HTTPS, HTTP redirection and HSTS with `includeSubdomain` and `Max-age` at around 180 days. This means the service must be visited within three months for ensuring HTTPS connection (before HSTS policy expired). The service, however, opts out of HSTS preload. By visiting the website, WEAPONS identified some of the website cookies (e.g., `JSESSIONID`) that do not set the `Secure` flag.

4) Krungsri Online: Krungsri Online is an internet banking service from Krungsri Bank [31]. The service provides HTTPS for security and privacy purposes. Nevertheless, our testing framework found a misconfiguration on the HSTS header, specifically, the header has empty content (Listing 2). Therefore, HSTS is ignored by browsers.

5) K-Cyber: K-Cyber is one of internet banking services from Kasikorn Bank [32]. Similar to the majority of services, K-Cyber offers HTTPS and HSTS with `includeSubdomains` and `Max-Age` of one year. Nonetheless, the service opts out of HSTS preload and some of the HTTP cookies do not have the `Secure` flag.

6) SCB Easy and KK e-Banking: SCB Easy is an internet banking website from Siam Commercial Bank [33]. KK e-Banking is from Kiatnakin Bank [34]. As opposed to the previous services, both websites only deploy HTTP, but HSTS. They are the only services in our testing that do not employ HSTS.

7) GSB Internet: Similar to other major services, Government Saving Bank [35] provides internet banking service on HTTPS with HSTS header with `Max-Age` of 180 days. However, it does not include the `includeSubdomains` directive. Therefore the HSTS policy would only apply to ib.gsb.or.th, but not to any subdomains under it. GSB Internet is one of the only two websites in our testing that have all cookies with the `Secure` flag.

8) Thanachart iNet: Thanachart iNet is an internet banking service for personal accounts from Thanachart Bank. The website has deployed both HTTPS and HSTS header with the preloaded option. However, after a closer inspection, none of the thanachartbank.co.th domains are on the preload list [23], therefore the domain will not be preloaded with supported browsers. The website has the longest `Max-Age` (two years) in our evaluation results.

B. Result Summary

HSTS and HSTS Preload. Due to criticality of these services, HSTS is highly recommended. Even though all websites we evaluated have deployed HTTPS, this does not imply that all requests are enforced to connect with only HTTPS as mentioned in Section II. Our results indicated that 7 out of 9 services have deployed HSTS, while only 2 services use HSTS preload. However, we found misconfigurations in 2 services, Krungsri Online and Thanachart iNet as mentioned in the previous Section. Most websites set up the `Max-Age` at the duration of 1 year and deploy `IncludeSubdomains` directive. With these misconfigurations, those websites are vulnerable to session hijacking attacks (e.g., SSL striping [3]).

HTTPS Redirection. As discussed in Section II, Configuring the appropriate redirect is necessary to offer a consistent and secure browsing experience. However, our results showed that only 4 services we audited have HTTPS redirection despite the fact that all have supported HTTPS. Since HSTS will be configured only on HTTPS, redirecting users to HTTPS as soon as possible is recommended.

Secure Cookie. We found that only 2 services have flagged all cookies `Secure`. As mentioned, `Secure` cookies are strongly recommended as it will ensure that the cookies are only sent through the encrypted channel. As presented in Section II these leaked cookies enable web tracking as well as potentially session hijacking [9], [36]–[39].

C. WEAPONS Performance Evaluation

Since WEAPONS can be run as online or offline service, website administrators and developers can deploy the offine version to avoid any delay from the network latency. In addition the offine version can be integrated as a part of their DevOps automation testing routines (e.g., Jenkins [40]) and ship the report for further investigations.

V. Related Work

The number of research has been presented in Section II. Here we provide a summary of works related to internet banking security in Thailand.

Kasemsan and Hunngam [41] presented a study of the internet banking security guideline model for the banking business in Thailand in 2005. The model is intended for ensuring the security of the internet banking services and raising security awareness. The paper presented several interesting concepts e.g., trust of the system, technology acceptance model, authentication. Nonetheless, it did not study the aspects of web encryption security. In 2007, Hamid et al. [42] compared numerous aspects between internet banking in Thailand and Malaysia including the security of the system from both sides. Their study revealed that the security systems of Thailand banks have a lower standard due to a higher of hacking incidents. However, their study did not include results from any technical aspects. A similar study had been proposed for measuring the quality of internet banking [43].

Subsorn and Limwiriyakul [44] evaluated internet banking security using a qualitative research method by comparing Australian and Thai commercial banks. Their analysis included e.g., encryption and digital certificate, logging, password requirements/restrictions, session management, two-factor authentication. Their work also provides some insights on web encryption. However, they studied web encryption in the aspects of the digital certificate, specifically size of the encryption key. They concluded that most of the selected Thai commercial banks have been deficient in providing internet banking security. Nevertheless, based on their study conducted in 2012, we have observed a significant improvement in internet banking in Thailand e.g., the deployment of HTTPS and HSTS as presented in Section IV

VI. Discussion

A. Existing online tools

Besides WEAPONS, there are other online tools that is already available to the public. The famous ones include, for example, SSL Labs by Qualys [45], SSL Checker [46], GeekFlare [47]. Similar to WEAPONS, given a website URL, these tools analyze various features of web encryption and mostly focus on in-depth analysis of SSL/TLS certificate e.g., supported versions, cipher suites. Nevertheless, none of these tools analyze secure HTTP cookie of all HTTPS requests on the webpage, HTTPS redirection. Some of these tools (e.g., SSL Labs) provide HSTS analysis. However, they do not cover the analysis of HSTS preload, max-age duration, and includeSubdomains directives. As most of these online tools analyze SSL/TLS certificates, we opt out to perform this analysis in this state of development.

B. Responsible Disclosure

To ensure the ethical nature of our research, we are planning to provide the details of our experiment to the bank that we found. At the time of writing this, we are working on an extended version of WEAPONS as well as in the process

of contacting and providing our results to those services. We believe that by shedding light on this significant privacy threat, we can incentivize the banking services to streamline support for ubiquitous encryption.

VII. Conclusion

In this paper, we presented our extensive study on the web encryption security of internet-banking websites in Thailand. We designed and implemented WEAPONS, an automated black-box web encryption analysis tool. WEAPONS navigates and collected a series of HTTP requests and responses data along with security mechanisms involved in web encryption. The system later performs an analysis and generates a report to the user for further investigations.

To this end, we found that all 9 internet-banking websites we audited have deployed HTTPS. 7 of those services have enforced encrypted connection by offering HSTS and HSTS preload. Nevertheless, we found a number of misconfigurations e.g., unsafe incorrect HSTS header and HTTP cookie setup which potentially lead to sensitive data exposure scenarios. We expect that the result generated from WEAPONS will be very useful to the developers for checking their web encryption deployment and therefore can help in reducing the chances of being exposed to information leakages and session hijacking attacks. We have made WEAPONS open-source so that the community can continue to build on it. All code and example results can be accessed at https://github.com/ssivakorn/weapons.

References

[1] NetScape. The SSL Protocol. https://web.archive.org/web/19970614020952/http://home.netscape.com/newsref/std/SSL.html.

[2] E. Butler, "Firesheep," 2010, http://codebutler.com/firesheep.

[3] M. Marlinspike, "New Tricks For Defeating SSL In Practice," *BlackHat DC*, Feb. 2009.

[4] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz, "The Web Never Forgets: Persistent Tracking Mechanisms in the Wild," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[5] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan, and E. W. Felten, "Cookies That Give You Away: The Surveillance Implications of Web Tracking," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15, 2015.

[6] C. Castelluccia, E. De Cristofaro, and D. Perito, "Private Information Disclosure from Web Searches," in *Privacy Enhancing Technologies*, ser. PETS '10, 2010.

[7] X. Zheng, J. Jiang, J. Liang, H. Duan, S. Chen, T. Wan, and N. Weaver, "Cookies Lack Integrity: Real-World Implications," in *Proceedings of the 24th USENIX Security Symposium*, 2015.

[8] A. Bortz, A. Barth, and A. Czeskis, "Origin cookies: Session integrity for web applications," in *Proceedings of the Web 2.0 Security and Privacy 2011 workshop*, ser. W2SP '11, 2011.

[9] S. Sivakorn, I. Polakis, and A. D. Keromytis, "The cracked cookie jar: Http cookie hijacking and the exposure of private information," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 724–742.

[10] Y. Liu, H. H. Song, I. Bermudez, A. Mislove, M. Baldi, and A. Tongaonkar, "Identifying Personal Information in Internet Traffic," in *Proceedings of the 3rd ACM Conference on Online Social Networks*, ser. COSN '15, 2015.

[11] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste, "The Cost of the "S" in HTTPS," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '14. ACM, 2014, pp. 133–140.

[12] M. Kranch and J. Bonneau, "Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning," in *Proceedings of the Network and Distributed System Security Symposium*, ser. NDSS '15, 2015.

[13] Carnegie Endowment for International Peace. Timeline of Cyber Incidents Involving Financial Institutions. https://carnegieendowment.org/specialprojects/protectingfinancialstability/timeline.

[14] Financial Times. Hacking ring linked to theft at Citibank ATMs. https://www.ft.com/content/75359ff0-8fff-11de-bc59-00144feabdc0.

[15] Capital One. Information on the Capital One Cyber Incident. https://www.capitalone.com/facts2019/.

[16] The New York Times. JPMorgan Chase Hacking Affects 76 Million Households. https://dealbook.nytimes.com/2014/10/02/jpmorgan-discovers-further-cyber-security-issues/.

[17] IETF. HTTP Over TLS. https://tools.ietf.org/html/rfc2818.

[18] Google. Secure your site with HTTPS. https://support.google.com/webmasters/answer/6073543.

[19] J. Hodges, C. Jackson, and A. Barth, "HTTP Strict Transport Security," RFC 6797, 2012.

[20] P. Chen, N. Nikiforakis, C. Huygens, and L. Desmet, "A dangerous mix: Large-scale analysis of mixed-content websites," in *Proceedings of the 16th Information Security Conference*, 2013.

[21] J. Selvi, "Bypassing HTTP Strict Transport Security," *BlackHat-EU*, 2014.

[22] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, , A. Pironti, and P.-Y. Strub, "Triple Handshakes and Cookie Cutters: Breaking and Fixing Authentication over TLS," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, 2014.

[23] Google. HSTS Preload. https://hstspreload.org/.

[24] IETF. RFC 6265. https://tools.ietf.org/html/rfc6265.

[25] S. Sivakorn, I. Polakis, and A. D. Keromytis, "I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs," in *IEEE European Symposium on Security and Privacy (EuroS&P) 2016*.

[26] SeleniumHQ Browser Automation. https://selenium.dev/.

[27] Selenium-Wire. https://pypi.org/project/selenium-wire/.

[28] Bangkok Bank. https://www.bangkokbank.com/en/Personal/Digital-Banking/Bualuang-iBanking.

[29] Krungthai Bank. https://www.ktb.co.th/en/personal.

[30] TMB Bank. https://www.tmbdirect.com/.

[31] Krungsri Bank. https://www.krungsrionline.com/.

[32] Kasikorn Bank. https://online.kasikornbankgroup.com/.

[33] Siam Commercial Bank. https://www.scbeasy.com/.

[34] Kiatnakin Bank. https://ebanking.kiatnakin.co.th/.

[35] Government Saving Bank. GSB Internet Banking. https://ib.gsb.or.th/retail/security/commonLogin.jsp.

[36] J. R. Mayer and J. C. Mitchell, "Third-Party Web Tracking: Policy and Technology," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 2012.

[37] S. Sivakorn, A. D. Keromytis, and J. Polakis, "That's the way the cookie crumbles: evaluating https enforcing mechanisms," in *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, 2016, pp. 71–81.

[38] S. Sivakorn, J. Polakis, and A. D. Keromytis, "Http cookie hijacking in the wild: Security and privacy implications," *Black Hat USA*, 2016.

[39] S. Sivakorn, "Understanding flaws in the deployment and implementation of web encryption," Ph.D. dissertation, Columbia University, 2018.

[40] Jenkins. https://www.jenkins.io/.

[41] K. Kasemsan and N. Hunngam, "Internet banking security guideline model for banking in thailand," *Communications of the IBIMA*, 2011.

[42] M. R. A. Hamid, H. Amin, S. Lada, and N. Ahmad, "A comparative analysis of internet banking in malaysia and thailand." *Journal of Internet Business*, no. 4, 2007.

[43] P. Leelapongprasut, P. Praneetpolgrang, and N. Paopun, "A quality study of internet banking in thailand," in *Proceedings of the 4th International Conference on eBusiness. Bangkok, Thailand*, 2005, pp. 6–1.

[44] P. Subsorn and S. Limwiriyakul, "A comparative analysis of internet banking security in thailand: A customer perspective," *Procedia engineering*, vol. 32, pp. 260–272, 2012.

[45] Qualys. SSL Server Test. https://www.ssllabs.com/ssltest/index.html.

[46] SSLChecker.com. https://www.sslchecker.com/sslchecker.

[47] GeekFlare. https://gf.dev/tests.