# การศึกษาเชิงประจักษ์ของการเข้ารหัสเว็บไซต์ธนาคารทางอินเทอร์เน็ตในประเทศไทย

# An Empirical Study of the Internet Banking Web Encryption in Thailand

**สุพรรณี ศิวากรณ์* ณัฐธยาน์ รุจิราธนาพัฒน์ พัชร์สิตา ศิรวงศ์ภัสสรา**
**สกุลชาย สารมาศ ยศภัทร เรืองไพศาล และ ชานนท์ ดวงพายัพ**

**Suphannee Sivakorn*, Nuttaya Rujiratanapat, Patsita Sirawongphatsara,**
**Sakulchai Saramat, Yotsapat Ruangpaisarn and Chanond Duangpayap**

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีราชมงคลตะวันออก
Department of Computer Science, Faculty of Science and Technology, Rajamangala University of Technology Tawan-ok
**Email: suphannee_si@rmutto.ac.th*

## บทคัดย่อ

ระบบธนาคารทางอินเทอร์เน็ตในประเทศไทยมีการพัฒนาไปอย่างรวดเร็ว สิ่งนี้ทำให้ความมั่นคงปลอดภัยทางออนไลน์มีความจำเป็น และเป็นที่ต้องการอย่างมากในปัจจุบัน และเนื่องจากความมั่นคงปลอดภัย และความเป็นส่วนตัวของผู้ใช้บนเว็บนั้นขึ้นอยู่กับ HTTPS ซึ่งเป็นโปรโตคอลการเข้ารหัสเว็บ เพื่อรักษาความมั่นคงปลอดภัยในการสื่อสารระหว่างผู้ใช้ และเว็บเซิร์ฟเวอร์ HTTPS จึงเป็นศูนย์กลางของระบบนิเวศของเว็บในปัจจุบัน แม้ว่าการใช้ HTTPS จะมีจำนวนเพิ่มขึ้น แต่ผลจากการศึกษาจำนวนมากแสดงให้เห็นว่าเว็บไซต์จำนวนมากนำ HTTPS ไปใช้อย่างไม่ถูกต้อง ทำให้ข้อมูลของผู้ใช้เสี่ยงต่อการรั่วไหล เช่น การดักฟังและโจมตีแบบคนกลาง (man-in-the-middle attack) การปรับใช้ HTTPS อย่างถูกต้องจึงมีความสำคัญมากสำหรับบริการธนาคารทางอินเทอร์เน็ต เนื่องจากมีข้อมูลที่ละเอียดอ่อนของผู้ใช้และเป็นเป้าหมายหลักสำหรับผู้ประกอบอาชญากรรม ในบทความนี้ ผู้วิจัยนำเสนอ WEAPONS ซึ่งเป็นเครื่องมือทดสอบกล่องดำแบบ offline เพื่อประเมินความสมบูรณ์ และความถูกต้องของการปรับใช้การเข้ารหัสเว็บ รวมถึงการปรับใช้ HTTPS และกลไกที่เกี่ยวข้องกับการเข้ารหัสเว็บ เช่น HSTS, คุกกี้ที่มีความปลอดภัย (Secure cookies), การเปลี่ยนเส้นทางเป็น HTTPS (HTTPS redirection), และการใช้ HSTS เป็นค่าเริ่มต้น (HSTS preload) โดยผู้วิจัยได้ใช้เครื่องมือดังกล่าวประเมินเว็บไซต์ธนาคารทางอินเทอร์เน็ตที่เป็นที่ยอดนิยมในประเทศไทย จำนวน 8 แห่ง ทั้งนี้จากผลการทดสอบแสดงให้เห็นว่า WEAPONS สามารถค้นหาข้อผิดพลาดในการปรับใช้ HTTPS ได้ และเปิดเผยจุดอ่อนที่อาจทำให้เกิดการโจมตีแบบคนกลาง รวมถึงจุดอ่อนอื่น ๆ ที่ผู้โจมตีอาจใช้ประโยชน์ในการเปิดเผยข้อมูลที่ละเอียดอ่อน

**คำสำคัญ :** HTTPS HSTS การเข้ารหัสเว็บ Web Encryption SSL/TLS WEAPONS

## 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 8 popular internet banking websites in Thailand. 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.

**Keywords :** HTTPS, HSTS, Web Encryption, SSL/TLS

## 1. บทนำ

การทำให้ผู้ใช้สามารถสื่อสารด้วยข้อมูลที่มีความอ่อนไหวทางออนไลน์ได้อย่างปลอดภัยเป็นรากฐานที่สำคัญของระบบนิเวศของเว็บในปัจจุบัน HTTPS เปิดตัวครั้งแรกในปี พ.ศ. 2537 โดย Netscape Communications[1] ถือเป็นจุดกำเนิดการเข้ารหัสเว็บ ซึ่งถูกออกแบบขึ้นมาเพื่อใช้ในการรักษาความมั่นคงปลอดภัยของอินเทอร์เน็ต ตัวอย่างเช่น การรับรองความถูกต้อง (Authentication) การถ่ายโอนข้อมูลอย่างปลอดภัย การดาวน์โหลดที่ปลอดภัย และการท่องเว็บแบบส่วนตัว เป็นต้น และเนื่องด้วยการอัปโหลดข้อมูลส่วนบุคคลจำนวนมากไปยังเว็บไซต์ที่เพิ่มขึ้นอย่างต่อเนื่องในชีวิตประจำวัน ทำให้ความเป็นส่วนตัวของผู้ใช้ในการสื่อสารแบบดิจิทัลถือเป็นเรื่องสำคัญและเร่งด่วน จากผลการศึกษางานวิจัยหลายชิ้นชี้ให้เห็นถึงความสำคัญของการรักษาการเชื่อมต่อเว็บให้ปลอดภัยจากผู้โจมตี โดยแสดงให้เห็นว่าผู้โจมตีสามารถเข้ายึด session และขโมยข้อมูลของผู้ใช้ได้ง่ายเพียงใด (Marlinspike, 2009; Acar et al., 2014; Castelluccia et al., 2010; Bortz et al., 2011; Sivakorn et al., 2016)

อย่างไรก็ตามเว็บไซต์ที่ให้บริการทางอินเทอร์เน็ตหลักจำนวนมากยังคงไม่ระมัดระวังในการให้บริการเนื้อหาผ่านการเชื่อมต่อที่ไม่ได้เข้ารหัส โดยไม่บังคับใช้การเชื่อมต่อที่เข้ารหัส (Englehardt et al., 2015; Castelluccia et al., 2010; Sivakorn et al., 2016; Liu et al., 2015) ทั้งนี้สืบเนื่องมาจากการกำหนดค่ากลไกความมั่นคงปลอดภัยที่เกี่ยวข้องอย่างไม่ถูกต้อง (Sivakorn et al., 2016; Kranch and Bonneau, 2015; Clark and van Oorschot, 2013) ซึ่งล้วนแต่ทำให้ผู้ใช้อาจถูกโจมตีโดยการขโมยข้อมูล โดยเฉพาะเว็บไซต์ที่มีความสำคัญยิ่งต้องมีการระมัดระวังการโจมตีลักษณะดังกล่าว เช่น เว็บไซต์ของธนาคาร เนื่องจากสถาบันทางการเงินมักเป็นศูนย์กลางของกิจกรรมที่มุ่งประสงค์ร้ายและฉ้อโกงของผู้ไม่หวังดี (Silver-Greenberg et al., 2014; Capital One, 2019) ธนาคารจึงต้องรับรองความถูกต้องของเทคโนโลยีและระบบความมั่นคงปลอดภัยของตนเอง เพื่อสร้างความมั่นใจให้กับลูกค้าอย่างต่อเนื่อง เมื่อประเทศไทยก้าวเข้าสู่ระบบการเงินบนอินเทอร์เน็ตอย่างเต็มรูปแบบ ความต้องการความมั่นคงปลอดภัยและความเป็นส่วนตัวทางออนไลน์จึงมีความจำเป็นอย่างมากในปัจจุบัน

ในบทความนี้ ผู้วิจัยได้นำเสนอและคิดค้นซอฟแวร์ชื่อ WEAPONS (Web Encryption Analysis Program for ONline Services) ซึ่งเป็นโปรแกรมวิเคราะห์การเข้ารหัสสำหรับเว็บที่ต้องการให้บริการออนไลน์ โดยโปรแกรมจะทำการทดสอบเพื่อประเมินความสมบูรณ์และความถูกต้องของการเข้ารหัสเว็บไซต์ เช่น HTTPS, การเปลี่ยนเส้นทางเป็น HTTPS (HTTPS redirection), HSTS และคุกกี้ที่ปลอดภัย (Secure cookies) โดยในการประเมินประสิทธิภาพของ WEAPONS ผู้วิจัยได้ใช้โปรแกรมดังกล่าวเพื่อวิเคราะห์และประเมินการเข้ารหัสของเว็บไซต์ธนาคารทางอินเทอร์เน็ตที่นิยมใช้ในประเทศไทย 8 แห่ง และยังได้พบการกำหนดค่าผิดพลาดจำนวนหนึ่งที่อาจนำไปสู่การเปิดเผยข้อมูลที่ละเอียดอ่อนได้

### (1) วัตถุประสงค์

โดยเป้าหมายของการวิจัยนี้มีสองประเด็นหลัก ประเด็นแรก คือ เพื่อเข้าใจสถานะการเข้ารหัสของเว็บไซต์ธนาคารทางอินเทอร์เน็ตของไทย ประเด็นที่สอง คือ เพื่อสร้างเครื่องมือทดสอบอัตโนมัติสำหรับประเมินการเข้ารหัสเว็บไซต์เพื่อระบุความผิดพลาดต่าง ๆ ซึ่งอาจนำไปสู่การเปิดเผยข้อมูลที่มีความละเอียดอ่อนของผู้ใช้

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

### (2) ที่มาและความสำคัญ

ในส่วนนี้ผู้วิจัยจะให้ข้อมูลสรุปเกี่ยวกับกลไกการรักษาความมั่นคงปลอดภัยและการเข้ารหัสเว็บและภาพรวมของการโจมตีที่ทำให้ผู้โจมตีสามารถเปิดเผยข้อมูลที่ละเอียดอ่อน

2.1 การเข้ารหัสเว็บ

มีโปรโตคอลและกลไกมาตรฐานจำนวนหนึ่ง ทั้งในฝั่งผู้ใช้งาน (web client หรือ web browser) และฝั่งเซิร์ฟเวอร์ เพื่อให้ผู้ใช้งานเว็บไซต์เกิดความมั่นใจในความปลอดภัยและความเป็นส่วนตัว

2.1.1 HTTPS: Hypertext Transfer Protocol Secure เป็นโปรโตคอลหลักในความมั่นคงปลอดภัยที่ออกแบบมาเป็นส่วนขยายของ HTTP เพื่อการสื่อสารที่มั่นคงปลอดภัย HTTPS หรือกล่าวคือเป็นการใช้งานการเข้ารหัสจากโปรโตคอล SSL/TLS บน HTTP[2] ซึ่งจะช่วยส่งข้อมูลโดยใช้โปรโตคอลที่เข้ารหัส (HTTPS) และให้ความมั่นคงปลอดภัยทาง HTTPS มักถูกตีความว่าเป็นการรักษาความมั่นคงปลอดภัยที่เป็นส่วนเสริมมากกว่า และมักไม่ถูกนำไปใช้สำหรับ request ทุก ๆ หน้าของเว็บไซต์ ถึงแม้ว่าจะมีคำแนะนำให้เว็บไซต์ใช้ HTTPS ในทุกหน้า ไม่ว่าเนื้อหาและความละเอียดอ่อนนั้นจะมากน้อยแค่ไหนก็ตาม[3] ทั้งนี้จากการสำรวจของ Google พบว่ามีจำนวน page ในอินเทอร์เน็ตที่ถูกเรียกผ่าน Google Chrome มากกว่า 95% ที่ได้ใช้ HTTPS (Google Transparency Report, 2022)

2.1.2 การเปลี่ยนเส้นทาง HTTPS (HTTPS Redirection): การพิมพ์ชื่อโดเมนที่ไม่มี https:// ลงในเบราว์เซอร์จะนำผู้ใช้ไปยัง HTTP โดยอัตโนมัติ ตัวอย่างเช่น การพิมพ์ example.com ลงในเบราว์เซอร์ นั้นเบราว์เซอร์จะทำการส่งคำขอ HTTP ไปยังเว็บเซิร์ฟเวอร์ example.com ก่อนจากนั้นจะขึ้นอยู่กับเว็บไซต์ที่จะเปลี่ยนเส้นทางเบราว์เซอร์ไปยัง HTTPS ซึ่งปกติจะทำโดยการเปลี่ยนเส้นทาง HTTP เช่น 301 Permanent, 302, 307 Temporary เพื่อให้เบราว์เซอร์ร้องขอในโปรโตคอล HTTPS แทน (https://example.com) ดังนั้น จึงเป็นหน้าที่ของเว็บเซิร์ฟเวอร์ในการเปลี่ยนเส้นทางเป็น HTTPS โดยเร็วที่สุดเพื่อความปลอดภัยของผู้ใช้ ซึ่งการเปลี่ยนเส้นทางโดย 301 Permanent เป็นคำแนะนำมาตรฐาน เนื่องจากเป็นการระบุให้เบราว์เซอร์ใช้การเปลี่ยนเส้นทาง HTTPS อย่างถาวร เพื่อหลีกเลี่ยงการรั่วไหลของข้อมูลให้มากที่สุด

2.1.3 HSTS: HTTP Strict Transport Security[4] เป็นส่วนหัวของ HTTP (HTTP header) การตอบสนอง HTTP ที่ใช้เป็นกลไกในการระบุให้เว็บไซต์แจ้งเว็บเบราว์เซอร์เพื่อให้เชื่อมต่อผ่าน HTTPS เท่านั้น โดยไม่ต้องอาศัยการเปลี่ยนเส้นทาง HTTPS จากฝั่งเซิร์ฟเวอร์ หรือจากกลไกอื่น ๆ เช่น HTTPS Everywhere[5] (Sivakorn et. al., 2016) หลักจากการการระบุนี้แล้ว เมื่อมีการเข้าถึงเว็บไซต์ เบราว์เซอร์จะรู้ว่าไซต์นั้นมี HTTPS และเปลี่ยนเส้นทางภายในไปยัง HTTPS ตามส่วนหัวของ Strict-Transport-Security ซึ่งหมายความว่าไม่มีการส่งข้อมูลผ่าน HTTP (ทุก ๆ request จะถูกส่งในรูปแบบการเข้ารหัส) ทำให้ไม่มีข้อมูลรั่วไหลระหว่างการส่ง Jackson and Barth (2008) ได้นำเสนอหลักการของกลไกนี้ครั้งแรก โดยเรียกกลไกดังกล่าวว่า ForceHTTPS ซึ่งเป็นส่วนขยายของเว็บเบราว์เซอร์ (Web browser extension) สำหรับการบังคับใช้การเชื่อมต่อแบบ HTTPS จนกลไกนี้ได้ถูกปรับปรุงและใช้สร้างเป็นมาตรฐาน HSTS ใน RFC 6797 หากไม่มี HSTS ผู้เข้าใช้งานเว็บไซต์ยังคงต้องสื่อสารกับเว็บไซต์ผ่าน HTTP (ไม่เข้ารหัส) อยู่ ตัวอย่างเช่น ก่อนการเปลี่ยนเส้นทางของ HTTPS หรือ คำขอในรูปแบบ HTTP ที่ซ่อนอยู่ในหน้าของเว็บไซต์ เช่น คำขอ HTTP จากเนื้อหาผสม (mixed content) (Chen et al., 2013) โดยการส่งขอมูลที่ไม่เข้ารหัสเหล่านี้ล้วนอาจสร้างโอกาสสำหรับการโจมตีแบบคนกลาง (man-in-the-middle attack) เช่น SSL Strip (Marlinspike, 2009) รวมถึงการโจมตีแบบการขโมย session (session hijacking) อื่น ๆ ซึ่งรายละเอียดการโจมตีที่เกี่ยวข้องสามารถดูได้ในส่วน 2.2

---

[2] IETF. RFC 2818 - HTTP Over TLS. https://tools.ietf.org/html/rfc2818.

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

[4] IETF. RFC 6797- HTTP Strict Transport Security. https://datatracker.ietf.org/doc/html/rfc6797.

[5] EFF. HTTPS Everywhere. https://www.eff.org/https-everywhere

อย่างไรก็ตาม มีงานวิจัยจำนวนหนึ่งที่ได้แสดงให้เห็นถึงสถานการณ์ที่ผู้โจมตีสามารถเลี่ยงกลไก HSTS ได้ โดยมีเหตุผลหลักคือเกิดจากนักพัฒนาเว็บไม่เข้าใจเกี่ยวกับการใช้กลไกเหล่านี้อย่างเหมาะสม โดยมักใช้ไปในทางที่ไม่ถูกต้องตามคำแนะนำ (Selvi, 2014; Bhargavan et al., 2014; Kranch and Bonneau, 2015; Bareño-Gutierrez et al., 2020)

2.1.4 การกำหนดค่า HSTS: การเปิดใช้งาน HSTS นั้น จะถูกกำหนดผ่านค่าส่วนหัวการตอบสนอง HTTP (HTTP response header) ทั้งนี้ การตั้งค่านี้จะต้องเป็นการตั้งค่าโดยการตอบสนองจาก HTTPS เท่านั้น (ไม่ใช่ HTTP) เนื่องจากผู้โจมตีอาจสกัดกั้นหรือแก้ไขหรือลบค่าดังกล่าวหากเป็นการตั้งค่าที่เกิดจากการเชื่อมต่อแบบ HTTP เพื่อให้เกิดความเข้าใจมากขึ้นเกี่ยวกับการประเมินที่ผู้วิจัยได้นำเสนอในส่วนที่ 4 ในที่นี้จะอธิบายคำสั่ง (HSTS directives) ที่ใช้ในโปรโตคอล HSTS และการกำหนดค่าที่ถูกต้องตามแนะนำ ดังนี้

*max-age*: เวลาเป็นวินาทีที่เบราว์เซอร์จะบังคับให้ไซต์เชื่อมต่อกับ HTTPS โดยต้องเป็นค่าจำนวนเต็มบวกใด ๆ และระยะเวลานี้จะได้รับการอัปเดตเมื่อมีการส่งส่วนหัว HSTS ใหม่ไปยังเบราว์เซอร์ อย่างไรก็ตาม ไม่มีคำแนะนำอย่างเป็นทางการสำหรับระยะเวลาที่ควรตั้งค่า max-age แต่ระยะเวลา max-age ที่ยาวขึ้นจะลดจำนวนการเชื่อมต่อ HTTP โดยการตั้งค่าเริ่มต้นที่พบบ่อยที่สุดคือระยะเวลาหนึ่งปี (max-age: 31536000)

*includeSubDomains*: เป็นหนึ่งในคำสั่งส่วนหัว HSTS ถูกตั้งค่าตามโดเมนย่อยและโดเมนที่พบใน URL ของเว็บไซต์ ตัวอย่างเช่น เมื่อผู้ใช้เข้าถึงเว็บไซต์ https://www.example.com/foo ด้วย HSTS ซึ่งจะถูกเปิดใช้งานบนหน้าใด ๆ ก็ตาม ที่เข้าถึงผ่าน www.example.com อย่างไรก็ตาม HSTS ไม่ได้เปิดใช้งานบนหน้าเว็บในโดเมนระดับบน (example.com) และหน้าอื่น ๆ ในโดเมนย่อย เช่น bar.www.example.com ด้วยคำสั่ง includeSubdomains เบราว์เซอร์จะเปิดใช้งาน HSTS บนโดเมนย่อยทั้งหมด ซึ่งเป็นหน้าใดก็ได้บน *.www.example.com ดังนั้น เพื่อความปลอดภัยจึงมีการแนะนำให้ใช้ includeSubdomains เนื่องจากจะบังคับใช้ HTTPS ทุกหน้าในโดเมนย่อย และหากเป็นไปได้ ควรตั้งค่า includeSubdomains ในระดับชื่อโดเมน (เช่น example.com) เพื่อป้องกันทุกหน้าบนชื่อโดเมน อย่างไรก็ตาม ค่า includeSubDomains ไม่ได้เป็นค่าบังคับและต้องมีตาม RFC 6797 หากแต่เป็นทางเลือกเพื่อเพิ่มความปลอดภัยอย่างสมบูรณ์

*preload*: เป็นหนึ่งในคำสั่งส่วนหัว HSTS ซึ่งควบคุมและบริหารโดย Google[6] โดยคำสั่ง preload HSTS จะถูกโหลดไว้ล่วงหน้าในเบราว์เซอร์ ดังนั้น เว็บไซต์ preload HSTS จะไม่เชื่อมต่อกับ HTTP เลยหากมีการใช้คำสั่งนี้อย่างถูกต้อง หากเว็บไซต์ต้องการอยู่ในรายการ preload เว็บไซต์จะต้องปฏิบัติตามข้อกำหนดในการส่งทั้งหมด รวมถึงมีใบรับรอง (Certificate) ที่ถูกต้อง การเปลี่ยนเส้นทาง HTTPS โดเมนย่อยของเว็บไซต์ทั้งหมดต้องอยู่บน HTTPS และส่วนหัว HSTS ที่มี max-age อย่างน้อยหนึ่งปี includeSubdomains และคำสั่ง preload ทั้งนี้ค่า preload ไม่ได้เป็นค่าบังคับและต้องมีตาม RFC 6797 หากแต่เป็นทางเลือกเพื่อเพิ่มความปลอดภัยอย่างสมบูรณ์

2.1.5 คุกกี้ที่ปลอดภัย (Secure cookie): คุกกี้ HTTP (HTTP cookie หรือ cookie) เป็นข้อมูลขนาดเล็กที่เซิร์ฟเวอร์ส่งไปยังเว็บเบราว์เซอร์ของผู้ใช้ ผ่านส่วนหัวของการตอบสนอง HTTP (HTTP response header) โดยใช้ส่วนหัวที่ชื่อว่า Set-Cookie[7] คุกกี้จะถูกจัดเก็บโดยเบราว์เซอร์ หลักจากการตั้งค่า คุกกี้จะถูกแนบไปกับคำขอ HTTP ในส่วนหัวที่ชื่อ Cookie มีการใช้คุกกี้ อย่างแพร่หลายเพื่อวัตถุประสงค์ในการตรวจสอบสิทธิ์ เนื่องจากสามารถใช้เป็น session ID ได้หลังจากผ่านกระบวนการตรวจสอบสิทธิ์ (authentication) ดังนั้น จะต้องมีการจัดเก็บและส่งคุกกี้ HTTP อย่างปลอดภัยเพื่อช่วยลดการโจมตีแบบขโมย session (session hijacking) โดยในมาตราฐาน RFC 6265 ได้มีกำหนดค่าสถานะที่ปลอดภัยสำหรับคุกกี้ (Secure attribute) โดยเป็นหนึ่งในตัวเลือกที่มีให้ ทั้งนี้คุกกี้ที่มีการตั้งค่าสถานะ Secure จะถูกส่งไปยังเว็บเซิร์ฟเวอร์ผ่าน HTTPS เท่านั้น ดังนั้น จึงมักมีคำแนะนำให้ใช้การตั้งค่าสถานะ Secure โดยเฉพาะในเว็บไซต์ที่มีเนื้อหาละเอียดอ่อนและการล็อกอิน

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

[7] IETF. RFC 6265 - HTTP State Management Mechanism. https://datatracker.ietf.org/doc/html/rfc6265.

2.2. การเปิดเผยข้อมูลที่ละเอียดอ่อน

การโจมตีแบบการขโมย Session (Session hijacking attack) เป็นหนึ่งในช่องทางหลักที่เปิดโอกาสให้ผู้โจมตีสามารถขโมยข้อมูลของผู้ใช้ได้ ในงานวิจัยชิ้นนี้ผู้วิจัยเน้นเฉพาะการโจมตีเกิดขึ้นผ่านการสื่อสารอันเนื่องมาจากความมั่นคงปลอดภัยของการเข้ารหัส ปัญหาด้านความมั่นคงปลอดภัยที่ศึกษาในงานวิจัยนี้ ส่วนใหญ่เป็นผลมาจากเว็บไซต์ที่ไม่บังคับใช้การเข้ารหัสในทุกหน้าและโดเมนย่อย เพื่อปกป้องคุกกี้ทั้งหมด โดยมีตัวอย่างการโจมตีที่เกิดขึ้นมากมายจากจุดอ่อนดังกล่าว เช่น Marlinspike ได้แสดงให้เห็นการโจมตีแบบ SSL Striping (Marlinspike, 2009) โดยเป็นการโจมตีที่อาศัยการเปลี่ยนแปลงคำขอ HTTP ครั้งแรกของผู้ใช้ไปยังเว็บเซิร์ฟเวอร์ (ก่อนที่จะเปลี่ยนเป็น HTTPS) นอกจากนี้ยังมีงานวิจัยที่แสดงให้เห็นถึงความเสี่ยงในการสนับสนุนเว็บไซต์ที่มีเนื้อหาผสม (mixed content) ระหว่าง HTTP และ HTTPS ซึ่งคือการมีหน้าเว็บที่เข้าถึงผ่าน HTTPS แต่มีเนื้อหาที่ต้องดึงข้อมูลผ่าน HTTP (Jackson and Barth, 2008) นอกเหนือจากนี้ Englehardt et al., 2015 อภิปรายว่าคุกกี้ HTTP ยังสามารถรั่วไหลผ่านการเชื่อมต่อ HTTP นั้น สามารถใช้สำหรับการติดตามการใช้งานของผู้ใช้จากบุคคลอื่น ๆ โดยการเชื่อมโยงคำขอที่แตกต่างกันไปยังผู้ใช้รายเดียวกัน ผลลัพธ์ของการศึกษานี้ เน้นย้ำถึงภัยคุกคามของการเชื่อมต่อที่ไม่ได้เข้ารหัส Liu et al., 2015 ศึกษาข้อมูลส่วนบุคคล (Personal Identiable Information) ที่ถูกส่งในเครือข่ายที่ไม่ได้เข้ารหัส นอกจากนี้ Sivakorn et al. (2016) แสดงให้เห็นถึงวิธีการในการใช้คุกกี้ เพื่อควบคุมระบบวิเคราะห์ความเสี่ยงขั้นสูงของ Google เพื่อใช้หลีกเลี่ยง Google reCAPTCHA โดยผู้โจมตีสามารถใช้คุกกี้ที่ขโมยมาได้เพื่อหลบเลี่ยงการตอบ CAPTCHA และ ยังศึกษาความไม่ปลอดภัยของเว็บไซต์สำคัญ เช่น Google, eBay และชี้ให้เห็นถึงภัยคุกคามต่อความเป็นส่วนตัวเนื่องจากการไม่บังคับใช้การเข้ารหัส และยังสามารถรวบรวมคุกกี้และข้อมูลผู้ใช้ส่วนใหญ่ผ่านเครือข่ายได้

2.3 ความมั่นคงปลอดภัยของธนาคารทางอินเทอร์เน็ตในประเทศไทย

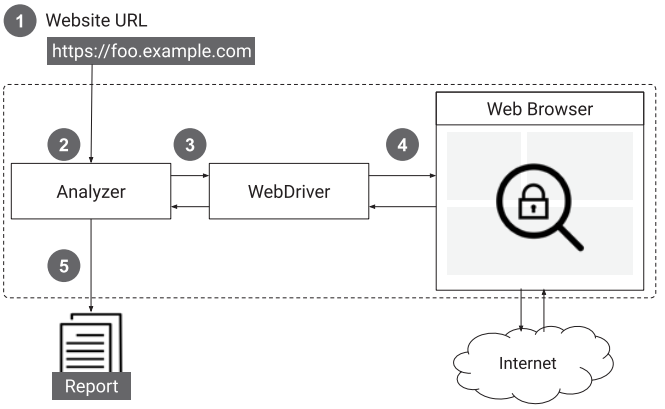
ในระยะเวลา 2-3 ปีที่ผ่านมาธนาคารทางอินเตอร์เน็ตในประเทศไทยเป็นเป้าหมายของการโจมตีของอาชญากรทางไซเบอร์และมีแนวโน้มจะเพิ่มสูงขึ้น (Posttoday, 2015; Bangkok Business, 2021; Prachachat Turakit, 2021) มีงานวิจัยจำนวนมากที่ทำการวิเคราะห์และเปรียบเทียบระบบธนาคารทางอินเทอร์เน็ตในประเทศไทย อย่างไรก็ตามงานวิจัยเหล่านั้นไม่ได้ศึกษาความมั่นคงปลอดภัยเชิงเทคนิค แต่เน้นการศึกษาด้านการประเมินความเสี่ยง ความไว้วางใจของผู้ใช้งาน (Namahoot et al., 2018; Aboobucker and Bao, 2018) และการประเมินความมั่นคงปลอดภัยผ่านจากการสำรวจข้อมูล (Khattak et al., 2021)

Kasemsan and Hunngam (2011) ได้นำเสนอคำแนะนำและแบบจำลองแนวทางการรักษาความมั่นคงปลอดภัยของธนาคารทางอินเทอร์เน็ตสำหรับธุรกิจธนาคารในประเทศไทย โดยนำเสนอแนวคิดที่น่าสนใจหลายประการ เช่น ความเชื่อถือของระบบ รูปแบบการยอมรับเทคโนโลยี การรับรองความถูกต้อง อย่างไรก็ตามผู้ศึกษาไม่ได้ทำการศึกษาแง่มุมของการรักษาความมั่นคงปลอดภัยในการเข้ารหัสเว็บ Hamid et al. (2010) ได้ศึกษาเปรียบเทียบระหว่างการรักษาความมั่นคงปลอดภัยของบริการธนาคารทางอินเทอร์เน็ตในประเทศไทยและมาเลเซีย โดยได้พบว่าระบบรักษาความมั่นคงปลอดภัยของธนาคารไทยมีมาตรฐานที่ต่ำกว่า เนื่องจากมีจำนวนเหตุการโจมตีที่สูงกว่า อย่างไรก็ตาม งานวิจัยนี้ไม่ได้รวมผลลัพธ์จากด้านเทคนิคใด ๆ Subsorn and Limwiriyakul (2012) ได้ประเมินความมั่นคงปลอดภัยธนาคารทางอินเทอร์เน็ตโดยการเปรียบเทียบธนาคารพาณิชย์ของประเทศออสเตรเลียและประเทศไทย ประกอบด้วยการวิเคราะห์การเข้ารหัสและใบรับรองดิจิทัล ขนาดของคีย์การเข้ารหัส และได้ข้อสรุปว่า ธนาคารพาณิชย์ของประเทศไทยนั้นที่ได้รับการคัดเลือกมาวิเคราะห์ส่วนใหญ่ยังขาดการรักษาความมั่นคงปลอดภัยในการให้บริการธนาคารทางอินเทอร์เน็ต อย่างไรก็ตาม เนื่องจากงานวิจัยนี้เป็นงานวิจัยจากปี พ.ศ. 2555 ปัจจุบันจะสังเกตเห็นได้ว่ามีการปรับปรุงระบบรักษาความมั่นคงปลอดภัยในบริการธนาคารทางอินเทอร์เน็ตในประเทศไทยไปมาก เช่น การใช้งาน HTTPS และ HSTS (Sivakorn et al., 2020)

## 2. วิธีการวิจัย

### ภาพรวมของระบบ

ในส่วนนี้จะอธิบายการออกแบบและการใช้งานแพลตฟอร์มการทดสอบ WEAPONS (Web Encryption Analysis Platform for Online Services) ที่ได้คิดค้นและใช้ในงานวิจัยนี้ โดยระบบได้พัฒนาขึ้นโดยใช้ Python 3, Library หลักที่ชื่อ Selenium[8] ซึ่งเป็นการควบคุมการทำงานอัตโนมัติของเบราว์เซอร์ และ Selenium-wire[9] ซึ่งเป็นส่วนขยายของ Selenium สำหรับการเข้าถึงคำขอ HTTP ผู้วิจัยเลือกใช้ Chromium WebDriver เพื่อให้สามารถใช้ประโยชน์จากฟังก์ชันการทำงานที่หลากหลายของเบราว์เซอร์ ที่มาจาก Google Chrome เวอร์ชั่น 90.0 โดยใช้ WebDriver ซึ่งสามารถใช้ automate ในการค้นหาองค์ประกอบ HTML, การดึงส่วนหัวของ HTTP ดังนั้น จึงสามารถออกแบบให้โปรแกรมสามารถอ่านค่าที่กำหนดจากเว็บไซต์ที่เกี่ยวข้องกับการเข้ารหัสเว็บได้หลายอย่าง เช่น HSTS, คุกกี้ HTTP และการเปลี่ยนเส้นทาง HTTPS ทั้งนี้ผู้ใช้สามารถปรับเปลี่ยน WebDriver ให้เป็นเวอร์ชั่นปัจจุบัน หรือเลือกใช้ WebDriver ของ Browser ชนิดอื่นได้เช่นกัน




**ภาพที่ 1 ภาพรวมของสถาปัตยกรรมหลักของระบบ WEAPONS**

ภาพที่ 1 นำเสนอภาพรวมของการทำงานของ WEAPONS เพื่อวิเคราะห์และตรวจสอบการเข้ารหัสเว็บของเว็บไซต์ WEAPONS ใช้ URL ของเว็บไซต์จากผู้ใช้ จากนั้นจึงทำการวิเคราะห์เว็บไซต์และส่งคืนผลลัพธ์ให้ผู้ใช้ โดยระบบประกอบด้วย 5 ขั้นตอนหลัก (1) การรับ URL เว็บไซต์ตามที่ผู้ใช้กำหนด (2) WEAPONS จะเริ่มการวิเคราะห์โดยการจำลองการใช้งานของผู้ใช้โดยเริ่มจากการส่งการสื่อสารกับ WebDriver ตามที่กล่าวไว้ข้างต้น (3) WebDriver จะเปิดเว็บไซต์ที่ต้องการทดสอบผ่าน Web Browser โดยทั้งหมดจะทำงานโดยอัตโนมัติ และรวบรวมผลลัพธ์ (4) Analyzer จะรวมรวมข้อมูลที่ได้ และวิเคราะห์ ค้นหาช่องโหว่ที่อาจเกิดขึ้นและการกำหนดค่าที่ไม่ถูกต้องในการเข้ารหัสเว็บตามที่อธิบายไว้ในส่วนที่ 2 ได้แก่ การใช้ HTTPS, การเปลี่ยนเส้นทาง HTTPS, HSTS, HSTS preload และการตั้งค่าสถานะคุกกี้ที่ปลอดภัย (Secure cookie) ขั้นสุดท้าย (5) ระบบจะส่งต่อสิ่งที่ค้นพบทั้งหมดกลับไปที่ได้ เพื่อเตรียมผลที่ผู้ใช้โดยการสร้างรายงาน HTML สำหรับการตรวจสอบเพิ่มเติม โดยกระบวนการทั้งหมดใช้เวลาประมาณ 60 วินาทีสำหรับแต่ละเว็บไซต์ อย่างไรก็ตาม ผู้ใช้ยังมีตัวเลือกเพื่อลดความล่าช้าในการรอโหลดหน้าเว็บจนเต็ม และสามารถเรียกใช้ WEAPONS พร้อมกันกับกระบวนการ WebDriver ที่แตกต่างกันบนเครื่องหรือเซิร์ฟเวอร์ทดลองเดียวกัน เพื่อให้การใช้เวลามีประสิทธิภาพมากขึ้น ซึ่งผู้วิจัยจะนำเสนอผลการใช้ WEAPONS เพื่อตรวจสอบเว็บไซต์ธนาคารทางอินเทอร์เน็ตในประเทศไทยในผลการวิจัยต่อไป

---

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

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

ทั้งนี้ หากผู้ที่สนใจสามารถเข้าทดลองใช้ WEAPONS ได้ที่ https://github.com/ssivakorn/WEAPONS พร้อมกับซอร์สโค้ด ตัวอย่างรายงาน คู่มือ และตัวอย่างการใช้งาน อีกทั้งผู้วิจัยยังอนุญาตให้ระบบดังกล่าวเป็นซอฟแวร์ในรูปแบบ Open-source เพื่อให้ผู้สนใจสามารถร่วมพัฒนาซอฟแวร์ดังกล่าวได้อีกด้วย

### การประเมินผล

ผู้วิจัยเลือกเว็บไซต์ธนาคารทางอินเทอร์เน็ตยอดนิยมในประเทศไทย ที่เปิดให้บริการบนอินเตอร์เน็ต จำนวน 8 แห่ง สำหรับประเมินซอฟแวร์ที่ได้พัฒนาขึ้น โดยดำเนินการในเดือนกรกฎาคม - สิงหาคม พ.ศ. 2564 โดยตารางที่ 1 แสดงรายการบริการที่เลือกพร้อมกับ URL บริการธนาคารทางอินเทอร์เน็ตที่ตรวจสอบ

**ตารางที่ 1 รายชื่อธนาคารทางอินเทอร์เน็ตและ URL ที่ให้บริการในประเทศไทยที่ได้รับการตรวจสอบ**

| บริการธุรกรรมทางอินเทอร์เน็ต | ธนาคาร | URL |
|---|---|---|
| Bualuang iBanking | กรุงเทพ | https://ibanking.bangkokbank.com |
| KTB NetBank | กรุงไทย | https://ktbnetbank.com |
| TTB Direct | ทีเอ็มบีธนชาต | https://ttbdirect.com |
| Krungsri Online | กรุงศรีอยุธยา | https://www.krungsrionline.com |
| K-Cyber | กสิกรไทย | https://online.kasikornbankgroup.com/K-Online/ |
| SCB Easy | ไทยพาณิชย์ | https://www.scbeasy.com/v1.4/site/presignon/index.asp |
| KKP e-Banking | เกียรตินาคินภัทร | https://bank.kkpfg.com |
| GSB Internet | ออมสิน | https://ib.gsb.or.th |

## 3. ผลการทดลองและวิจารณ์ผล

ในส่วนนี้เป็นการให้รายละเอียดเกี่ยวกับผลของการศึกษาการตรวจสอบบริการธนาคารทางอินเทอร์เน็ต ผู้วิจัยเลือกบริการเหล่านี้จากธนาคารที่ได้รับความนิยมที่สุดในประเทศไทย โดยพบว่า ยังมีเว็บไซต์ของธนาคารบางแห่งที่มีการกำหนดค่าผิดพลาดในการใช้ HTTPS ซึ่งอาจทำให้ข้อมูลรั่วไหลได้ ตารางที่ 2 นำเสนอภาพรวมของบริการธนาคารทางอินเทอร์เน็ตและผลลัพธ์ของงานวิจัย ดังต่อไปนี้และเพื่อความเข้าใจเราจะสรุปผลของการตรวจสอบของแต่ละเว็บไซต์ในส่วนต่อไป ทั้งนี้เพื่อให้การศึกษาเป็นไปอย่างปลอดภัยและป้องกันผลกระทบที่อาจจะเกิดขึ้น ผู้วิจัยได้กำหนดรายชื่อสมมติและไม่เรียงลำดับ (ธนาคาร A – H) เพื่อแทนการใช้ชื่อธนาคารและบริการจริง โดยสามารถติดต่อผู้วิจัยเพื่อขอรายละเอียดและผลการวิจัยตามชื่อธนาคารและชื่อบริการจริง

3.1 ธนาคาร A: เป็นธนาคารที่ให้บริการธนาคารทางอินเทอร์เน็ตลำดับต้นๆ ในประเทศไทย จากการทดสอบพบว่า บริการดังกล่าวไม่ได้มีการอนุญาตให้เชื่อมต่อผ่าน HTTP และรองรับเฉพาะ HTTPS รวมถึง HSTS ด้วย IncludeSubdomains และ max-age ในระยะเวลาหนึ่งปี แต่ไม่ได้ปรับใช้ HSTS preload อย่างไรก็ตาม เครื่องมือ WEAPONS ได้ตรวจพบว่ายังมีคุกกี้บางส่วนที่ยังไม่ได้ตั้งค่าให้เป็น Secure

3.2 ธนาคาร B: ผลการทดสอบพบว่าธนาคาร B เป็นเพียงเว็บไซต์เดียวที่มีการปรับใช้ HTTPS ที่สมบูรณ์ที่สุด ซึ่งรวมถึง HTTPS, การเปลี่ยนเส้นทาง HTTPS, HSTS, includeSubdomains, max-age ในระยะเวลาหนึ่งปี และ HSTS preload บริการนี้เป็นเพียงหนึ่งในสองเว็บไซต์ธนาคารทางอินเทอร์เน็ตที่ได้รับตรวจสอบว่าได้ปรับใช้ HSTS preload จากการตรวจสอบเพิ่มเติม ธนาคาร B มีการปรับใช้ HSTS บนชื่อโดเมน รวมถึงโดเมนย่อยทั้งหมด (“include_subdomains”: true) ที่สามารถพบใน Preload List[10] ในการให้บริการ ส่งผลให้โดเมนย่อยทั้งหมดรวมถึงหน้าเข้าสู่ระบบ เชื่อมต่อกับ HTTPS บนเบราว์เซอร์ที่รองรับเสมอ นอกจาก HTTPS และ HSTS แล้ว คุกกี้ทั้งหมดที่พบเมื่อโหลดหน้าเข้าสู่ระบบจะถูกตั้งค่าสถานะความปลอดภัย ดังที่กล่าวไว้ ซึ่งทำให้มั่นใจได้ว่าคุกกี้ที่ละเอียดอ่อนจะไม่ถูกส่งออกไปผ่านการเชื่อมต่อ HTTP

[10] Google. HSTS Preload List. https://source.chromium.org/chromium/chromium/src/+/master:net/http/transport_security_state_static.json

**ตารางที่ 2 สรุปผลการตรวจสอบเว็บไซต์ธนาคารทางอินเทอร์เน็ตในประเทศไทย**

| บริการธุรกรรมทางอินเทอร์เน็ต | HTTPS | การเปลี่ยนเส้นทางเป็น HTTPS (HTTPS Redirection) | HSTS | | | | สถานะคุกกี้ที่ปลอดภัย (Secure Cookie) |
|---|---|---|---|---|---|---|---|
| | | | ส่วนหัว (Header) | Include Subdomains | Max-Age | Preload | |
| ธนาคาร A | Yes | - | Yes | Yes | 31536000 | No | Mixed |
| ธนาคาร B | Yes | Yes (302) | Yes | Yes | 31536000 | Yes | Full |
| ธนาคาร C | Yes | Yes (302) | Yes | Yes | 16070400 | No | Mixed |
| ธนาคาร D | Yes | - | Yes | No | N/A | No | Mixed |
| ธนาคาร E | Yes | - | Yes | Yes | 31536000 | No | Mixed |
| ธนาคาร F | Yes | Yes (301) | No | - | - | - | Full |
| ธนาคาร G | Yes | Yes (301) | Yes | - | - | - | Mixed |
| ธนาคาร H | Yes | Yes (302) | Yes | No | 15552000 | No | Full |

Yes: พบการเปิดใช้งาน • No: ไม่พบการเปิดใช้งาน
Mixed: พบการใช้บางส่วน • Full พบการใช้ทั้งหมด

3.3 ธนาคาร C: ธนาคาร C ให้บริการผ่านอินเทอร์เน็ตโดยเว็บไซต์ให้บริการผ่าน HTTPS, การเปลี่ยนเส้นทาง HTTP และ HSTS พร้อมด้วยโดเมนย่อยและ max-age ประมาณ 180 วัน ซึ่งหมายความว่าต้องเข้าใช้บริการภายในสามเดือนเพื่อให้แน่ใจว่ามีการเชื่อมต่อ HTTPS (ก่อนที่นโยบาย HSTS จะหมดอายุ) นอกจากนี้ WEAPONS สามารถตรวจพบคุกกี้บางส่วนของเว็บไซต์ (เช่น JSESSIONID) ที่ไม่ได้ตั้งค่าสถานะความมั่นคงปลอดภัย (Secure)

3.4 ธนาคาร D: จากการตรวจสอบพบว่ามี HTTPS เพื่อความมั่นคงปลอดภัยและความเป็นส่วนตัว และไม่ให้บริการทาง HTTP อย่างไรก็ตาม จากกรอบการทดสอบที่ใช้ พบว่ามีการกำหนดค่าที่ไม่ถูกต้องในส่วนหัว HSTS โดยเฉพาะอย่างยิ่ง ส่วนหัว (HSTS Header) ที่มีเนื้อหาว่างเปล่า (strict-transport-security:) ดังนั้น เบราว์เซอร์จึงไม่ตั้งค่า HSTS ในกรณีนี้

3.5 ธนาคาร E: เช่นเดียวกับบริการของธนาคารในประเทศไทยส่วนใหญ่ ธนาคาร E ได้ให้บริการผ่าน HTTPS และ HSTS พร้อมโดเมนย่อยและ max-age ระยะเวลาหนึ่งปี อย่างไรก็ตาม คุกกี้ HTTP บางตัวที่ไม่ได้ตั้งค่าสถานะความมั่นคงปลอดภัย

3.6 ธนาคาร F: จากผลการทดสอบพบว่าบริการธนาคารออนไลน์ของธนาคาร F เป็นเว็บไซต์เดียวที่จากการทดสอบที่ไม่ได้มีการเปิดใช้งาน HSTS ดังนั้นจึงมีระดับความปลอดภัยที่ต่ำกว่าธนาคารอื่น ๆ เนื่องจากอาจถูกโจมตีโดยการขโมย Session หรือ SSL Strip ตามที่ได้กล่าวมาใน 2.2

3.7 ธนาคาร G: เป็นเว็บไซต์ธนาคารทางอินเทอร์เน็ตจากธนาคาร G โดยพบว่าได้มีรองรับการใช้งานแบบ HTTPS เท่านั้น เว็บไซต์ได้มีความพยายามในการกำหนดค่า HSTS ในส่วนหัว แต่ได้กำหนดค่าไว้ไม่ถูกต้องโดยใช้ “stricttransport-security: max-age=expireTime” แทนมาตรฐานที่ได้กล่าวมาในส่วนที่ 2.1.3 ดังนั้น การใช้งาน HSTS ของเว็บไซต์ดังกล่าวจึงไม่มีผล และอาจถูกโจมตีในรูปแบบที่ได้กล่าวมาเช่นกัน

**ตารางที่ 3 แสดงการตั้งค่า HSTS สองครั้งของธนาคาร H**

| | |
|---|---|
| 1 | HTTP/1.1 200 OK |
| 2 | X-Frame-Options: SAMEORIGIN |
| 3 | Strict-Transport-Security: max-age=31536000; includeSubDomains; preload |
| 4 | Cache-Control: no-cache="set-cookie, set-cookie2" |
| 5 | Connection: close |
| 6 | Content-Type: text/html;charset=UTF-8 |
| 7 | Content-Language: en-US |
| 8 | Transfer-Encoding: chunked |
| 9 | Strict-Transport-Security: max-age=15552000 |

3.8 ธนาคาร H: เช่นเดียวกับบริการหลักของธนาคารอื่น ๆ ธนาคาร H ให้บริการธนาคารทางอินเทอร์เน็ตบน HTTPS พร้อมส่วนหัว HSTS โดยมี Max-Age 180 วัน และมีการตั้งค่า includeSubdomains ด้วยเช่นกัน อย่างไรก็ดี ในการทดสอบ เราได้พบการตั้งค่าที่ผิดพลาดจากเว็บไซต์เนื่องจากพบการตั้งค่า "Strict-Transport-Security" ถึงสองครั้งใน Header (ตารางที่ 3 ในบรรทัดที่ 3 และ 9) โดยพบว่าค่าที่ไม่เท่ากันคือ ค่า max-age ได้แก่ 31536000 (1 ปี) และ 15552000 (180 วัน) รวมถึงค่า includeSubdomains และ Preload ที่ไม่มีการตั้งค่าในส่วนหัวที่สอง ทั้งนี้ เมื่อนำค่า domain ของธนาคารดังกล่าวไปค้นหาใน Preload List ตามได้กล่าวไว้ พบว่า domain ดังกล่าวไม่ได้อยู่ใน Preload List ตามที่ได้ตั้งค่าไว้

## 4. สรุปผลการศึกษา

4.1 HSTS และ HSTS Preload: แม้ว่าเว็บไซต์ทั้งหมดที่ทำการประเมินมีการปรับใช้ HTTPS แล้ว แต่ไม่ได้หมายความว่า คำขอทั้งหมดถูกบังคับใช้เพื่อเชื่อมต่อกับ HTTPS เท่านั้นตามที่กล่าวไว้ในส่วนที่ 2 ผลลัพธ์ของการทดสอบระบุว่าบริการของธนาคารทางอินเทอร์เน็ต 7 ใน 8 แห่ง ได้ปรับใช้ HSTS ในขณะที่มีเพียง 1 แห่งเท่านั้นที่ใช้ HSTS Preload อย่างไรก็ตาม พบว่า ธนาคาร D และธนาคาร G มีการกำหนดค่าผิดตามที่กล่าวไว้ในส่วนที่แล้ว และธนาคารออมสินที่มีการตั้งค่า HSTS ที่ซ้ำซ้อนและไม่ตรงกัน การศึกษายังพบว่าเว็บไซต์ส่วนใหญ่ตั้งค่า Max-Age ในระยะเวลา 1 ปี และโดยส่วนใหญ่ได้มีที่การใช้คำสั่ง IncludeSubdomains ร่วมด้วย

4.2 การเปลี่ยนเส้นทาง HTTPS: การกำหนดค่าการเปลี่ยนเส้นทางที่เหมาะสมเป็นสิ่งจำเป็นเพื่อมอบประสบการณ์การท่องเว็บที่มีความเสถียรและมั่นคงปลอดภัย ถึงแม้ว่าในปัจจุบัน Chrome ตั้งแต่เวอร์ชั่น 90 จะกำหนดให้มีการใช้ HTTPS เป็นค่าพื้นฐานในการเรียกเว็บแล้วก็ตาม (Chromium Blog, 2021) อย่างไรก็ดีผู้ใช้ที่ยังใช้เวอร์ชันที่เก่าและโปรแกรมอื่น ๆ ยังจำเป็นที่ต้องการเปลี่ยนเส้นทางเป็น HTTPS จากผลจากการวิจัยแสดงให้เห็นว่ามีเว็บไซต์ธนาคารทางอินเทอร์เน็ตนั้นได้ที่มีการเปลี่ยนเส้นทาง HTTPS โดยได้มีการเปลี่ยนเส้นทางผ่านการใช้ 301 และ 302 redirection หรือเลือกที่จะไม่รอบรับการเชื่อมต่อผ่าน HTTP เลย

4.3 คุกกี้ที่ปลอดภัย: จากการศึกษาวิจัยพบว่ามีเว็บไซต์ธนาคารทางอินเทอร์เน็ตเพียง 3 แห่งเท่านั้นที่ตั้งค่าสถานะความมั่นคงปลอดภัยของคุกกี้ทั้งหมด การให้ใช้คุกกี้ที่ปลอดภัย จะช่วยให้แน่ใจว่าคุกกี้จะถูกส่งผ่านช่องทางที่เข้ารหัสเท่านั้น

## 5. บทสรุป

งานวิจัยชิ้นนี้เป็นการนำเสนอการศึกษาเกี่ยวกับการรักษาความมั่นคงปลอดภัยการเข้ารหัสของเว็บไซต์ธนาคารทางอินเทอร์เน็ตในประเทศไทยระหว่างปี พ.ศ. 2564 โดยผู้วิจัยได้ออกแบบและใช้งาน WEAPONS ซึ่งเป็นเครื่องมือวิเคราะห์การเข้ารหัสเว็บแบบอัตโนมัติ โดย WEAPONS จะทำหน้าที่นำทางและรวบรวมชุดของคำขอ HTTP และ HTTPS ของเว็บไซต์ที่ต้องการทดสอบ พร้อมเก็บข้อมูลที่เกี่ยวข้องกับกลไกความมั่นคงปลอดภัยที่เกี่ยวข้องกับการเข้ารหัสเว็บ

ได้แก่ HSTS การตั้งค่าคุกกี้ เป็นต้น จากนั้นระบบจะทำการวิเคราะห์และสร้างรายงานให้ผู้ใช้ตรวจสอบต่อไป

ในการประเมิน ผู้วิจัยใช้ WEAPONS เพื่อตรวจสอบเว็บไซต์ในเว็บไซต์ธนาคารออนไลน์ของประเทศไทย โดยเลือกบริการธนาคารทางอินเทอร์เน็ตที่สำคัญและได้รับความนิยม จากผลการวิจัยพบว่าเว็บไซต์ธนาคารทางอินเทอร์เน็ตทั้ง 8 แห่งที่ตรวจสอบ มีการปรับใช้ HTTPS และบริการของธนาคารทางอินเทอร์เน็ต 6 ใน 8 แห่ง มีการบังคับใช้การเชื่อมต่อที่เข้ารหัสโดยเสนอ HSTS และ เพียงหนึ่งแห่ง ที่ได้ใช้ HSTS preload นอกจากนี้เครื่องมือดังกล่าวยังสามารถค้นพบว่าบริการส่วนใหญ่ยังคงมีการตั้งค่าคุกกี้ HTTP ที่ไม่ปลอดภัย และยังสามารถตรวจพบการตั้งค่า HSTS ที่ผิดพลาด ทั้งนี้เพื่อให้การศึกษาเป็นไปอย่างปลอดภัยและป้องกันผลกระทบที่อาจจะเกิดขึ้น ผู้วิจัยได้กำหนดรายชื่อสมมติ โดยไม่เรียงลำดับ (ธนาคาร A – H) เพื่อใช้แทนชื่อจริงของธนาคารและบริการ โดยสามารถติดต่อผู้วิจัยเพื่อขอรายละเอียดและผลการวิจัยตามชื่อจริงของธนาคารและบริการดังกล่าว

ผู้วิจัยคาดหวังว่าผลลัพธ์ที่สร้างจาก WEAPONS จะมีประโยชน์มากสำหรับนักพัฒนาในการตรวจสอบการปรับใช้การเข้ารหัสเว็บ โดยสามารถช่วยลดโอกาสรั่วไหลของข้อมูลและการโจมตีแบบการขโมย session ได้ ผู้วิจัยได้สร้าง WEAPONS เป็น open-source ขึ้นเพื่อให้สามารถร่วมกันพัฒนาต่อไปได้ และได้นำเสนอชุดคำสั่งและตัวอย่างการใช้งานไว้ที่ https://github.com/ssivakorn/weapons

## 6. เอกสารอ้างอิง


Aboobucker, I. and Y. Bao. 2018. **What obstruct customer acceptance of internet banking? Security and privacy, risk, trust and website usability and the role of moderators**. The Journal of High Technology Management Research. 29(1), 109-123.

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

Bangkok Business. 2021. **Attack on mobile banking in ASEAN, up 60%!! 5 Pinning Methods for Minimizing Transactions**. https://www.bangkokbiznews.com/tech/960865. Accessed August 27, 2021.

Bareño-Gutierrez, R., A. López-Sevillano, F. Piraquive and R. Gonzalez Crespo. **2020. Analysis of WEB browsers of HSTS security under a man attack in the MITM environment**. Proceedings of the 5th International Conference on Knowledge Management in Organisations.

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

Bortz, A., A. Barth and A. Czeskis. 2011. **Origin cookies: Session integrity for web applications**. Proceedings of the Web 2.0 Security and Privacy 2011 Workshop.

Capital One. 2019. **Information on the Capital One Cyber Incident**. Retrieved August 27, 2022, from https://www.capitalone.com/facts2019/

Castelluccia, C., E. De Cristofaro and D. Perito. 2010. **Private Information Disclosure from Web Searches**. Privacy Enhancing Technologies.

Chen, P., N. Nikiforakis, C. Huygens and L. Desmet. 2013. **A Dangerous Mix: Large-scale Analysis of Mixed-content Websites**. Proceedings of the 16th Information Security Conference.

Chromium Blog. 2021. **A safer default for navigation: HTTPS**. https://blog.chromium.org/2021/03/a-safer-default-for-navigation-https.html. Accessed August 27, 2022.

Clark, J. and P.C. van Oorschot. 2013. **SoK: SSL and HTTPS: Revisiting Past Challenges and Evaluating Certificate Trust Model Enhancements**. Proceedings of the 2013 IEEE Symposium on Security and Privacy.

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

Google Transparency Report. 2022. **HTTPS encryption on the web -- HTTPS Encryption by Chrome platform**. https://transparencyreport.google.com/https/overview?hl=en. Accessed August 27, 2022.

Hamid, R., A. Hamid, H. Amin, S. Lada and N. Ahmad. 2010. **A Comparative Analysis of Internet Banking in Malaysia and Thailand**. Journal of Internet Business. No. 4.

Jackson, C. and A. Barth. 2008. **ForceHTTPS: Protecting High-security Web Sites from Network Attacks**. Proceedings of the 17th International World Wide Web Conference.

Kasemsan, K. and N. Hunngam. 2011. **Internet Banking Security Guideline Model for Banking in Thailand**. Communications of the IBIMA. DOI: 10.5171/2011.787725.

Khattak, S., S. Jan, I. Ahmad, Z. Wadud and F.Q. Khan. 2021. **An Effective Security Assessment Approach for Internet Banking Services via Deep Analysis of Multimedia Data**. Multimedia Systems 27: 733-751.

Kranch, M. and J. Bonneau. 2015. Upgrading HTTPS in Mid-Air: **An Empirical Study of Strict Transport Security and Key Pinning**. Proceedings of the Network and Distributed System Security Symposium.

Leelapongprasut, P., P. Praneetpolgrang and N. Paopun. 2005. **A Quality Study of Internet Banking in Thailand**. Proceedings of the 4th International Conference on eBusiness.

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

Marlinspike, M. 2009. **New Tricks For Defeating SSL In Practice**. BlackHat USA.

Namahoot, K. S., and T. Laohavichien. 2018. **Assessing the Intentions to Use Internet Banking: The Role of Perceived Risk and Trust as Mediating Factors**. International Journal of Bank Marketing. 36(2): 256-276.

Posttoday. 2015. **Four banks prepare to deal with hackers**. https://www.posttoday.com/finance-stock/news/396588. Accessed August 27, 2021.

PrachachatTurakit. 2021. **Launching the trend of cybersecurity in Thailand in the year 65 "Government and banks" are the most targeted attacks**. https://www.prachachat.net/ict/news-826506. Accessed August 27, 2021.

Selvi, J. 2014. **Bypassing HTTP Strict Transport Security**. BlackHat EU.

Silver-Greenberg, J., M. Goldstein and N. Perlroth. 2014. **The New York Times. JPMorgan Chase Hacking Affects 76 Million Households**. https://dealbook.nytimes.com/2014/10/02/jpmorgandiscovers-further-cyber-security-issues/. Accessed August 1, 2021.

Sivakorn, S., I. Polakis and A.D. Keromytis. 2016. **The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information**. Proceedings of the 2016 IEEE Symposium on Security and Privacy: 724-742.

Sivakorn, S., I. Polakis and A.D. Keromytis. 2016. **I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs**. Proceedings of the 1st IEEE European Symposium on Security and Privacy.

Sivakorn, S., A.D. Keromytis and I. Polakis. 2016. **That's the Way the Cookie Crumbles: Evaluating HTTPS Enforcing Mechanisms**. Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society: 71-81.

Sivakorn, S., P. Sirawongphatsara and N. Rujiratanapat. 2020. **Web Encryption Analysis of Internet Banking Websites in Thailand**. Proceedings of 17th International Joint Conference on Computer Science and Software Engineering: 139-144.

Subsorn, P. and S. Limwiriyakul. 2012. **A Comparative Analysis of Internet Banking Security in Thailand: A Customer Perspective**. Procedia Engineering, 32: 260-272.

(Received: 29/Aug/2021, Revised: 24/Jun/2022, Accepted: 27/Jun/2022)