วารสารวิชาการพระจอมเกล้าพระนครเหนือ ปีที่ 33 ฉบับที่ 2 เม.ย.–มิ.ย. 2566
The Journal of KMUTNB., Vol. 33, No. 2, Apr.–Jun. 2023

บทความวิจัย

การวิเคราะห์ปัญหาการทำงานผิดพลาดของกลไกเอชเอสทีเอสและการจู่โจมด้วยการเปลี่ยนเอสเอสแอล

ภารเดช คะเชนรัมย์ ดรุณี พ่วงพรพิทักษ์ และ สมนึก พ่วงพรพิทักษ์*
กลุ่มวิจัยความมั่นคงสารสารสนเทศและเครือข่ายขั้นสูง คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
เอกชัย พ่วงพรพิทักษ์
กรมสอบสวนคดีพิเศษ กระทรวงยุติธรรม

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 08 9453 2159 อีเมล: somnuk.p@msu.ac.th DOI: 10.14416/j.kmutnb.2021.07.007
รับเมื่อ 9 มีนาคม 2564 แก้ไขเมื่อ 8 มิถุนายน 2564 ตอบรับเมื่อ 17 มิถุนายน 2564 เผยแพร่ออนไลน์ 29 กรกฎาคม 2564

บทคัดย่อ

การโจมตีด้วยการเปลี่ยนเอสเอสแอลเป็นหนึ่งในเทคนิคที่รู้จักกันอย่างแพร่หลาย เพื่อโจมตีเว็บไซต์ที่ใช้เอชทีทีพีเอส ดังนั้น กลไกเอชเอสทีเอสจึงได้ถูกนำเสนอและใช้งานเพื่อสยบการโจมตีดังกล่าว แต่อย่างไรก็ตาม จากการศึกษาเมื่อไม่นานมานี้ หลายงาน ได้แสดงให้ว่าการโจมตีด้วยการเปลี่ยนเอสเอสแอลเดิมสามารถนำมาใช้โจมตีระบบธนาคารออนไลน์ และเว็บอีคอมเมิร์ซได้ผลอีกครั้งแม้มีการตั้งค่าเอชเอสทีเอสแล้วก็ตาม ดังนั้นงานวิจัยนี้จึงทำการตรวจสอบ และวิเคราะห์หาเหตุผลเบื้องหลังการทำงานล้มเหลวของกลไกเอชเอสทีเอส และการกลับมาโจมตีได้ใหม่ของการโจมตีด้วยการเปลี่ยนเอสเอสแอล เพื่อวิเคราะห์ปัญหาได้ทำการทดลองบนเครือข่ายเพื่อการทดสอบต่อเว็บธนาคารออนไลน์ของไทย 11 ธนาคาร ระบบเว็บอีคอมเมิร์ซ 4 เว็บ และเว็บอาสาสมัครอีก 2 เว็บ และยังมีการวิเคราะห์เอชทีทีพีเรสพอนซีพเฮดเดอร์ และวิเคราะห์สคริปต์ที่แฮกเกอร์ใช้ในการจู่โจม ในที่สุดสาเหตุของปัญหาก็ได้รับการวิเคราะห์และแนวทางในแก้ปัญหาได้ถูกเสนอแนะ

คำสำคัญ: กลไกเอชเอสทีเอส การจู่โจมด้วยการเปลี่ยนเอสเอสแอล ความมั่นคงเว็บ

การอ้างอิงบทความ: ภารเดช คะเชนรัมย์, ดรุณี พ่วงพรพิทักษ์, สมนึก พ่วงพรพิทักษ์ และ เอกชัย พ่วงพรพิทักษ์, "การวิเคราะห์ปัญหาการทำงานผิดพลาดของกลไกเอชเอสทีเอสและการจู่โจมด้วยการเปลี่ยนเอสเอสแอล," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 33, ฉบับที่ 2, หน้า 626–636, เม.ย.–มิ.ย. 2566.

Research Article

Problem Analysis of HSTS Malfunction and SSL Stripping Attack

Paradet Khachenrum, Darunee Puangpronpitag and Somnuk Puangpronpitag*
Information Security & Advanced Network (ISAN) Research Group, Faculty of Informatics, Mahasarakham University, Maha Sarakham, Thailand
Egachai Puangpronpitag
Department of Special Investigation (DSI), Ministry of Justices, Bangkok, Thailand

* Corresponding Author, Tel. 08 9453 2159, E–mail: somnuk.p@msu.ac.th DOI: 10.14416/j.kmutnb.2021.07.007
Received 9 March 2021; Revised 8 June 2021; Accepted 17 June 2021; Published online: 29 July 2021

Abstract

SSL stripping attack was one of the most notorious techniques to hack HTTPS websites. So, HTTP Strict Transport Security (HSTS) mechanism had been proposed and deployed to subdue the attack. However, a few recent studies have shown that the old SSL stripping attack could be deployed to effectively attack several on-line banking and e-commerce web sites again even with HSTS configuration. Hence, this paper investigates and analyzes reasons behind the malfunction of HSTS and the return of SSL stripping attacks. To analyze the problem, testbed experiments on 11 Thai online banking, 4 e-commerce websites and 2 volunteer websites, an analysis of HTTP response headers and hacker's scripts are done. The cause of problems has finally been analyzed and the solutions are suggested.

Keywords: HTTP Strict Transport Security (HSTS) Mechanism, SSL Stripping Attack, Web Security

Please cite this article as: P. Khachenrum, D. Puangpronpitag, S. Puangpronpitag, and E. Puangpronpitag, "Problem analysis of HSTS malfunction and SSL stripping attack," *The Journal of KMUTNB*, vol. 33, no. 2, pp. 626–636, Apr.–Jun. 2023 (in Thai).

1. บทนำ

การให้บริการต่างๆ ผ่านเว็บไซต์มีประเด็นที่ต้องให้ความสำคัญ คือ ข้อมูลที่สื่อสารกันระหว่างผู้ใช้งานกับผู้ให้บริการ หรือเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์จำเป็นต้องถูกเข้ารหัสเพื่อป้องกันการถูกดักจับข้อมูล โดยในปัจจุบันเทคโนโลยีสำคัญที่ใช้ คือ HTTPS (Hypertext Transfer Protocol Secure) [1] ซึ่งอาศัยโพรโทคอล HTTP ร่วมกับโพรโทคอล TLS (Transport Layer Security) [2] เพื่อปกป้องข้อมูลระหว่างการสื่อสาร ซึ่ง HTTPS จะอาศัยโครงสร้างพื้นฐานกุญแจสาธารณะ (Public Key Infrastructure: PKI) [3] การเข้ารหัสแบบสมมาตร (Symmetric) และอสมมาตร (Asymmetric) ทำให้ข้อมูลอยู่ในรูปแบบ Cipher Text ที่สามารถป้องกันการถูกดักจับได้

อย่างไรก็ตาม ได้มีเทคนิคที่ใช้โจมตี HTTPS ถูกพัฒนาขึ้นโดยเฉพาะอย่างยิ่งการโจมตีด้วยการเปลือยเอสเอสแอล (SSL Stripping Attack) ซึ่งถูกเสนอโดย Marlinspike และคณะ [4] ตั้งแต่ ค.ศ 2009 โดยจะทำให้การทำงานของโพรโทคอล HTTPS ถูกเปลี่ยนไปเป็น HTTP ที่ไม่มีการเข้ารหัส ผู้โจมตีจึงสามารถดักจับข้อมูลสำคัญ เช่น ชื่อผู้ใช้ และรหัสผ่านได้โดยเทคนิคการโจมตีนี้ มีมิจฉาชีพทั่วโลกรวมถึงในประเทศไทย ได้นำไปใช้ในการก่ออาชญากรรมเพื่อโจมตีระบบธนาคารออนไลน์ (Internet Banking) ระบบการค้าอิเล็กทรอนิกส์ (E-Commerce) และบริการอื่นๆ ที่สำคัญ

ดังนั้น SSL Stripping Attack จึงถือเป็นภัยคุกคามร้ายแรงที่สร้างปัญหาให้กับบริการผ่านเว็บไซต์มายาวนาน มีหลายงานวิจัยได้เสนอแนวทางแก้ไขแต่จากการศึกษาพบว่ายังขาดประสิทธิภาพในการป้องกัน เช่น SSLock [5] พบปัญหาในมาตราฐานการพัฒนาระบบและมีความยุ่งยากในการปรับใช้กับเว็บไซต์ HProxy [6] และ HTTPSLock [7] ทำได้เพียงตรวจสอบการโจมตีไม่อาจป้องกันได้ ระบบ ISAN-HTTPS Enforcer [8] เป็น JavaScript API สำหรับเว็บเซิร์ฟเวอร์เพื่อบังคับการสื่อสารให้เป็น HTTPS ที่ฝั่ง Client โดยเสมือนผู้ใช้พิมพ์ https:// เข้าไปเอง ซึ่งสามารถป้องกันการโจมตีด้วย SSL Stripping Attack ได้ แต่จากงานวิจัย [9] พบว่า ยังถูกโจมตีได้ด้วยการปรับ Python Script ของ SSL Stripping Attack เพื่อปลด JavaScript Tag ของระบบป้องกันนี้ออกได้ และระบบ Click2Enforce [10] เป็น Extension ของ Google Chrome ที่ให้ผู้ใช้เพิ่ม Domain Name ที่ต้องการลงใน List เพื่อบังคับใช้ HTTPS ในการสื่อสารในภายหลัง แต่วิธีนี้ต้องอาศัยความร่วมมือจากผู้ใช้งานมากเกินกว่าที่จะเป็นไปได้โดยทั่วไป ซึ่งหากผู้ใช้งานมีความระวังอยู่แล้ว ก็ย่อมสังเกตตัวสัญลักษณ์กุญแจล็อกของ HTTPS ได้เอง ดังนั้น SSL Stripping Attack ก็ย่อมไม่ใช่ปัญหากับกลุ่มผู้ใช้ดังกล่าว

ปัจจุบันกลไกที่ถูกเสนอให้ใช้ในการป้องกัน SSL Stripping Attack คือ กลไก HSTS (HTTP Strict Transport Security) [11] และยังเป็นหนึ่งในมาตรฐานของ IETF ตามเอกสาร RFC 6797 ซึ่งกลไก HSTS ทำหน้าที่เป็นกลไกส่วนเสริมของโพรโทคอล HTTPS ที่เปิดให้เว็บเซิร์ฟเวอร์ “บังคับ” ให้เว็บเบราว์เซอร์เชื่อมต่อผ่าน HTTPS เท่านั้น แม้ผู้ใช้จะไม่ระบุว่าต้องการใช้ HTTPS ก็ตาม ทำให้เว็บไซต์ที่ต้องการความมั่นคงสูง เช่น ระบบธนาคารออนไลน์ ระบบการค้าอิเล็กทรอนิกส์ มีการปรับใช้อย่างแพร่หลาย

ด้วยการตั้งค่าเพื่อใช้กลไก HSTS ของหลายเว็บไซต์ โดยเฉพาะอย่างยิ่งธนาคารออนไลน์ในประเทศไทย ทำให้ปัญหาการโจมตีด้วย SSL Stripping Attack ดูจะสิ้นสุดลงแล้ว กระทั่งเมื่อเดือนตุลาคม พ.ศ 2562 มีหลายกลุ่มวิจัย เช่น [12] ได้ทำการวิเคราะห์ปัญหาความมั่นคงของเว็บไซต์ทั้งในและนอกประเทศไทย โดยสำรวจเว็บไซต์ที่ให้บริการธนาคารออนไลน์ รวมถึงระบบเซอร์วิสที่สำคัญต่างๆ พบว่า ระบบเว็บไซต์หลายแห่ง โดยเฉพาะอย่างยิ่งธนาคารออนไลน์ ที่ดูเหมือนจะได้รับการป้องกันด้วยกลไก HSTS ไปแล้ว แต่กลับถูกโจมตีได้อีกครั้งทั้งสคริปต์การจูโจมแบบใหม่ของแฮกเกอร์ที่มีการเผยแพร่ และวิธีการโจมตีด้วยสคริปต์เดิมของ Moxie Marlinspike (ที่ไม่น่าจะได้ผลแล้ว) ดังนั้น SSL Stripping Attack จึงยังกลับมาเป็นภัยคุกคามต่อความมั่นคงของบริการผ่านเว็บไซต์ แม้จะมีการตั้งค่ากลไก HSTS เพื่อป้องกันตามเอกสารต่างๆ [13], [14] ที่มีการเผยแพร่ในอินเทอร์เน็ต

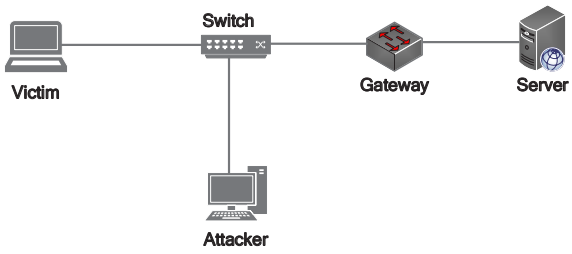
งานวิจัยนี้เป็นความร่วมมือระหว่างทีมผู้วิจัยกับฝ่ายคดีเทคโนโลยีสารสนเทศ กรมสอบสวนคดีพิเศษ กระทรวงยุติธรรม โดยมีจุดมุ่งหมายดังนี้ 1) ตรวจสอบเว็บไซต์ธนาคาร

ออนไลน์ในประเทศว่ายังถูกโจมตีด้วยการเปลือยเอสเอสแอลอยู่หรือไม่? 2) วิเคราะห์ปัญหาความผิดปกติ/ความล้มเหลวของกลไก HSTS และวิเคราะห์การโจมตีของแฮกเกอร์ โดยอาศัย SSL Stripping Attack ด้วยสคริปต์การโจมตีแบบเดิม และใหม่ เพื่อให้เข้าใจสาเหตุที่ SSL Stripping Attack กลับมาโจมตีได้ใหม่อีกครั้ง และ 3) เพื่อเสนอแนวทางการแก้ไขปัญหาป้องกันการถูกโจมตีดังกล่าว โดยผลลัพธ์จากงานวิจัยนี้ จะเป็นส่วนช่วยป้องกันอาชญากรรมทางคอมพิวเตอร์ต่อเว็บไซต์ที่ให้บริการสำคัญต่างๆ ทั้งระบบธนาคารออนไลน์ ระบบการค้าอิเล็กทรอนิกส์ และระบบเซอร์วิสที่สำคัญที่ให้บริการประชาชนอื่นๆ ซึ่งมีความสำคัญต่อเศรษฐกิจ สังคม และการบริหารประเทศในยุคดิจิทัล

2. วัสดุ อุปกรณ์และวิธีการวิจัย

2.1 แรงจูงใจของงานวิจัยนี้

SSL Stripping Attack เป็นภัยคุกคามร้ายแรงที่สร้างปัญหาให้กับระบบเว็บไซต์มายาวนาน มีหลายงานวิจัยที่พยายามเข้ามาแก้ไขปัญหาดังกล่าว จนกระทั่ง HSTS ได้ถูกเลือกใช้เป็นมาตรฐานในการป้องกัน และดูเหมือนว่าปัญหาจะจบลงแล้ว แต่จากการตรวจสอบของหลายงานวิจัย เช่น [12] ในช่วง พ.ศ 2562 พบว่า การโจมตีด้วย SSL Stripping Attack ผ่าน Bettercap Script [15] สามารถโจมตีเว็บไซต์ธนาคารออนไลน์ในประเทศไทยหลายแห่งได้ แม้เว็บไซต์ดังกล่าวมีการปรับใช้กลไก HSTS ในการบังคับการเชื่อมต่อ HTTPS ก็ตาม นอกจากนี้ SSL Stripping Scripts แบบเดิมของ Marlinspike ที่เคยจู่โจมไม่ได้ผลจากการตั้งค่า HSTS ตามคำแนะนำของหลายแหล่ง เช่น [8], [9] ก็กลับมาโจมตีได้ผลอีกครั้ง กอปรกับคดีที่มีการโจมตีระบบธนาคารออนไลน์ได้เกิดขึ้นในประเทศไทยหลายคดี ในช่วงหลายปีที่ผ่านมามีบางคดีได้ถูกกำหนดให้เป็นคดีพิเศษ ทั้งนี้เพราะมูลค่าความเสียหาย ความซับซ้อนของทั้งเทคนิควิธีในการโจมตี และวิธีทางกฎหมายในการสืบสวนสอบสวนคดี ดังนั้น ฝ่ายคดีเทคโนโลยีสารสารสนเทศ กรมสอบสวนคดีพิเศษ กระทรวงยุติธรรม จึงได้ร่วมมือกับทีมวิจัยเพื่อวิเคราะห์ปัญหาดังกล่าวอย่างละเอียด เพื่อเข้าใจปัญหา และเสนอวิธีการแก้ไขทั้งทางเทคนิควิธี และทางกฎหมาย โดยงานวิจัยนี้เป็นส่วนหนึ่งของความร่วมมือดังกล่าวที่มองด้านเทคนิควิธี



รูปที่ 1 Experimental Environment

2.2 เครื่องมือที่ใช้ในการวิจัย

เครื่องเหยื่อ (Victim) ใช้ CPU Intel i5 RAM 8GB โดยมี MS Windows 10 เป็นระบบปฏิบัติการ และใช้ Google Chrome เป็นเว็บเบราว์เซอร์ เครื่องของผู้โจมตี (Attacker) ใช้ CPU Intel i5 RAM 8GB โดยใช้ Kali Linux 2020.1 เป็นระบบปฏิบัติการ พร้อมติดตั้งโปรแกรม Bettercap, Ettercap, ARP Spoofing Tools และสคริปต์ในการจู่โจม SSL/TLS ของ Hacker แบบใหม่ที่แพร่หลายในอินเทอร์เน็ตในช่วง ค.ศ. 2019 และสคริปต์ SSL Strip Attack เดิมของ Marlinspike เพื่อใช้ในการโจมตี

การทดลองทำบน Test-bed มีสภาพแวดล้อมในการทดลองแสดงดังรูปที่ 1 ทั้งแบบ Wireless และ Wired Networks โดยเครื่องผู้โจมตี และเครื่องเหยื่ออยู่ในวง LAN เดียวกัน เครื่องเหยื่อใช้ Google Chrome Browser เพื่อเข้าสู่หน้า Login ของเว็บไซต์ปลายทาง และเครื่องผู้โจมตีทำการแทรกกลางการสื่อสาร ด้วยเทคนิค ARP Poisoning และทำการ SSL Stripping Attack ด้วยสคริปต์ 2 แบบ คือ แบบใหม่ที่ Hacker สร้างขึ้นและแจกใน ค.ศ. 2019 และแบบเดิมของ Marlinspike ที่มีมาตั้งแต่ ค.ศ. 2009 และดักจับข้อมูลด้วย Wireshark

2.3 เว็บที่ใช้ในการทดลอง

เว็บไซต์ที่นำมาทดลองประกอบด้วย เว็บไซต์ธนาคารออนไลน์ในประเทศไทย จำนวน 11 เว็บ เว็บไซต์ที่ให้บริการ

ระบบ E-Commerce 4 เว็บ เว็บไซต์ที่อาสาสมัครร่วมทดสอบ 2 เว็บ (โดยเป็นเว็บไซต์ของกลุ่มวิจัย และทีวีออนไลน์) โดยเว็บไซต์ธนาคารในประเทศไทยทั้ง 11 เว็บ เป็นการเลือกแบบเจาะจงตามความต้องการของกรมสอบสวนคดีพิเศษ เพื่อทราบสถานะภาพความมั่นคงปลอดภัยของธนาคารไทย ณ ปัจจุบันต่อปัญหาภัยคุกคามดังกล่าว และเว็บ E-Commerce ทั้ง 4 เว็บ ก็ถูกเลือกอย่างเจาะจง เนื่องจากเป็นเว็บไซต์ที่มีลูกค้าเป็นคนไทยจำนวนมาก ซึ่งเป็นไปตามความต้องการของกรมสอบสวนคดีพิเศษเช่นกัน เพื่อทราบสถานะการณ์ความมั่นคงปลอดภัยและใช้เป็นข้อมูลในการหาแนวทางการป้องปรามคดีที่จะเกิดในประเทศไทยต่อไป สำหรับเว็บอาสาสมัคร 2 เว็บ ที่ร่วมทดสอบเป็นเว็บของพันธมิตรของกลุ่มวิจัยที่เจาะจงนำมาใช้ทดสอบ เพราะมีความพยายามตั้งค่าเพื่อป้องกันขั้นสูงและใส่กลไกการป้องกันเพิ่มเติม ซึ่งจะสามารถนำมาวิเคราะห์เปรียบเทียบเพื่อเรียนรู้ปัญหา

ทั้งนี้การทดลองกระทำภายใต้การกำกับของนิติกรจากส่วนคดีเทคโนโลยีสารสนเทศ กรมสอบสวนคดีพิเศษ เพื่อไม่ให้ผิดจรรยาบรรณและกฎหมาย โดยไม่มีวัตถุประสงค์เจาะเข้าไปในระบบผู้ให้บริการ หรือก่อให้เกิดผลกระทบต่อการทำงานของระบบผู้ให้บริการแต่อย่างใด เพียงแต่เป็นการทดลองดักจับข้อมูลระหว่างการสื่อสารเท่านั้น โดยใช้เหยื่อที่สมมุติขึ้นและข้อมูลชื่อผู้ใช้และรหัสผ่านที่ทำการทดสอบเป็นของผู้ทดลองเอง หรือเป็นเพียงค่าปลอมที่ตั้งขึ้นที่ไม่ใช่ของจริง

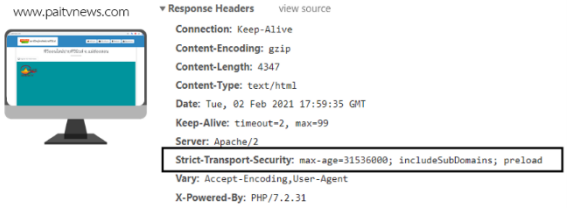
2.4 การวิเคราะห์กลไก HSTS

นอกจากการทดลอง SSL Stripping Attack ในกลุ่มตัวอย่างเว็บไซต์แล้ว เพื่อเข้าใจปัญหาอย่างแท้จริง งานวิจัยนี้ยังทำการ 1) ตรวจสอบและวิเคราะห์ HTTP Response Header ของเว็บไซต์ที่นำมาทดลองว่ามีการตั้งค่ากลไก HSTS หรือไม่อย่างไร 2) ตรวจสอบและวิเคราะห์ HSTS Preload List และ 3) ตรวจสอบและวิเคราะห์ Hacker Scripts ที่ใช้ในการโจมตี

3. ผลการทดลอง

3.1 ผลการตรวจสอบการตั้งค่ากลไก HSTS

การตรวจสอบการตั้งค่ากลไก HSTS ของเว็บไซต์ที่ใช้ในการทดลองทั้ง 17 เว็บไซต์ สามารถทำได้โดยการ Request เว็บไซต์ผ่านเว็บเบราว์เซอร์ จากนั้น Inspect Network แล้วเลือกดูข้อมูล Response Headers ก็จะพบ HSTS Header ดังแสดงรูปที่ 2



รูปที่ 2 ตัวอย่าง HSTS Config. ใน HTTP Header

ตารางที่ 1 ผลการตั้งค่า HSTS ของเว็บธนาคารในไทย

เว็บไซต์*	HSTS		
	Header	Max-Age	Preload
A	Yes	31536000	No
B	Yes	31536000	Yes
C	Yes	31536000	No
D	Yes	31536000	No
E	Yes	16070400	No
F	Yes	15552000	No
G	Yes	No	No
H	ไม่พบ HSTS config.		No
I	ไม่พบ HSTS config.		No
J	ไม่พบ HSTS config.		No
K	ไม่พบ HSTS config.		No

* เพื่อสงวนชื่อธนาคาร จึงใช้อักษรย่อแทนชื่อของธนาคาร

จุดประสงค์ของการตรวจสอบค่า HTTP Header นี้เพื่อดูว่ามีเว็บไซต์ที่หน่วยงานของรัฐต้องการให้สำรวจ ทั้งธนาคารออนไลน์ และ E-Commerce มีการตั้งค่า HSTS เพื่อป้องกันการโจมตีหรือไม่ และตั้งค่าเหมาะสมหรือไม่?

จากตารางที่ 1 หากดูจาก HTTP Header พบว่า 11 เว็บไซต์ของธนาคารออนไลน์ในไทยที่ทดสอบ มี 4 เว็บ ที่ใน Header ไม่พบการตั้งค่า HSTS ซึ่งน่าจะทำให้โดน SSL Stripping Attack ได้โดยง่าย เห็นได้ว่าทั้ง 4 ธนาคาร ในไทยไม่มีการปรับปรุงกลไกการป้องกันโจมตีตามมาตรฐานขั้นต้น ส่วนอีก 3

เว็บธนาคารมีการตั้งค่า HSTS แต่ค่า Max Age Configuration ไม่เหมาะสม (ค่าที่แนะนำโดย Google คือ 31536000 วินาทีขึ้นไป [16]) ซึ่งน่าจะโดนโจมตีได้ และมีเพียง 1 ธนาคาร ที่เหมือนจะตั้งค่า HSTS เป็นแบบ Preload (ที่น่าจะเหมาะสมที่สุด ซึ่งจะได้แสดงผลต่อไป)

ตารางที่ 2 ผลการตั้งค่า HSTS ของเว็บ E-Commerce

เว็บไซต์*	HSTS		
	Header	Max-Age	Preload
L	Yes	47474747	Yes
M	Yes	31536000	No
N	Yes	31536000	No
O	Yes	31536000	No

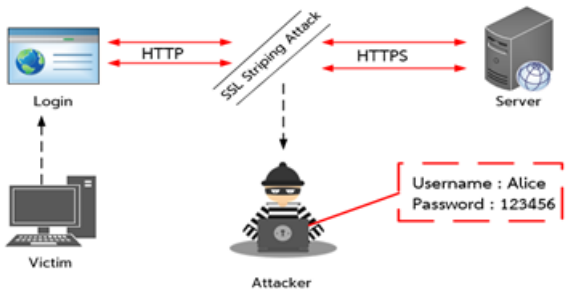
* เพื่อสงวนชื่อ E-commerce Websites จึงใช้อักษรย่อแทน

จากตารางที่ 2 หากดูจาก HTTP Header พบว่า 4 E-Commerce Web Sites มีการตั้งค่า HSTS ทั้ง 4 เว็บ และมีค่า Max-Age ที่เหมาะสม โดยมีเพียง 1 เว็บ คือ L ที่ตั้งค่า HSTS เป็นแบบ Preload (ที่น่าจะเหมาะสมที่สุด)

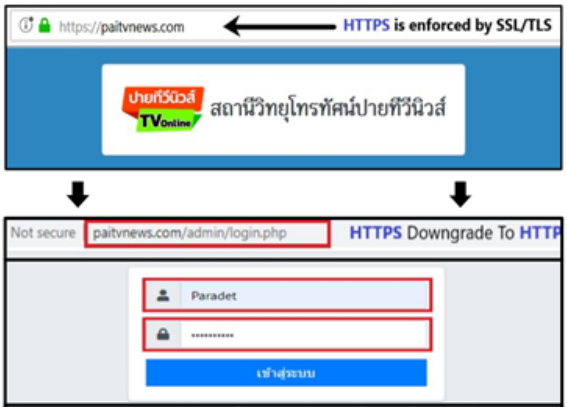
ตารางที่ 3 ผลการตั้งค่า HSTS ของเว็บที่อาสาให้ทดสอบ

เว็บไซต์	HSTS		
	Header	Max-Age	Preload
isanmsu.com	ไม่พบ HSTS config.		No
paitvnews.com	Yes	31536000	Yes

จากตารางที่ 3 หากดูจาก HTTP Header ใน 2 เว็บที่อาสาให้ทดสอบพบว่า isanmsu.com ใน Response Header ไม่มีการตั้งค่า HSTS config และน่าจะถูก SSL Strip Attack ได้ ส่วน paitvnews.com พบมีการตั้งค่า HSTS เป็นแบบ Preload (ที่น่าจะเหมาะสมที่สุด) และไม่น่าจะถูก SSL Strip Attack ได้ ในขั้นตอนนี้การวิเคราะห์ในเบื้องต้นจะดูเหมือนว่า isanmsu.com ไม่มั่นคงปลอดภัย แต่ paitvnews.com มั่นคงปลอดภัย ซึ่ง 2 เว็บอาสาสมัครนี้ เป็นส่วนที่ร่วมมือกับทีมวิจัย เพื่อจะได้นำไปสู่การวิเคราะห์ในขั้นตอนต่อไปว่าการอ่านค่า Response Header ไม่ใช่วิธีการวิเคราะห์ที่ถูกต้องสมบูรณ์ (ซึ่งจะได้กล่าวต่อไป)



รูปที่ 3 รูปแบบการโจมตีด้วย SSL Stripping Attack



รูปที่ 4 ตัวอย่างผลการ Strip และ Sniff

3.2 ผลการทดลองโจมตีด้วย SSL Stripping Attack

เพื่อให้ทราบว่าการโจมตีด้วย SSL Stripping Attack ต่อกลุ่มตัวอย่างเว็บไซต์ ทั้ง 17 เว็บ มีผลเป็นอย่างไร เป็นไปตามความคาดหวังหลังอ่านค่า HSTS Response Header หรือไม่ จึงได้ทำการทดลอง Strip และดักจับข้อมูล (Sniff) ในหน้า Login เพื่อเก็บชื่อผู้ใช้และรหัสผ่านโดยจำลองการโจมตีบน Test-bed แสดงดังรูปที่ 3 และตัวอย่างผลการโจมตีจริงของเว็บไซต์ paitvnews.com แสดงดังรูปที่ 4

ในการทดลองได้ใช้เครื่องมือโจมตีทั้งสองแบบ คือ SSL Strip Script ดั้งเดิมของ Marlinspike หรือ Ettercap และ Bettercap Script ใหม่ที่ Hacker มีการเผยแพร่ในช่วงประมาณ ค.ศ. 2019 ผลปรากฏว่าทั้งสองสคริปต์ให้ผลลัพธ์ของการทดลองเหมือนกัน ดังแสดงในตารางที่ 4–6

ตารางที่ 4 ผลการโจมตีเว็บไซต์ธนาคารออนไลน์ในไทย

เว็บไซต์*	HSTS		SSL Strip Attack	
	Max-Age	Preload	SSL Strip	Sniff
A	31536000	No	✓	✓
B	31536000	Yes	×	×
C	31536000	No	✓	✓
D	31536000	No	✓	✓
E	16070400	No	✓	×
F	15552000	No	✓	✓
G	Yes	No	✓	✓
H	ไม่พบ HSTS config.		✓	✓
I	ไม่พบ HSTS config.		✓	✓
J	ไม่พบ HSTS config.		✓	✓
K	ไม่พบ HSTS config.		✓	✓

* เพื่อสงวนชื่อธนาคาร จึงใช้อักษรย่อแทนชื่อของธนาคาร

จากตารางที่ 4 จะเห็นว่ามีเพียง 1 ธนาคาร ที่ตั้งค่า HSTS เป็นแบบ Preload ที่ไม่โดน SSL Strip ทำให้ไม่สามารถ Sniff ข้อมูลได้ และมี 10 ใน 11 ธนาคาร ที่หน้าเว็บโดน SSL Stripping Attack ปลด HTTPS ไปเป็น HTTP ได้ โดยใน 10 ธนาคาร ที่ถูก Strip ได้นี้มี 9 ธนาคาร ที่ถูก Sniff รหัสผ่านได้โดยง่าย ดังตัวอย่างแสดงในรูปที่ 5 ทั้งนี้มี 1 ธนาคาร ที่แม้ป้องกัน SSL Strip Attack ไม่ได้ แต่จากการตรวจสอบพบว่า มีกลไกการ Hash Password ทำให้แม้จะโดน Sniff ก็ไม่รู้ว่ารหัสผ่านคืออะไร

ตารางที่ 5 ผลการโจมตีเว็บไซต์ E-Commerce

เว็บไซต์*	HSTS		SSL Strip Attack	
	Max-Age	Preload	SSL Strip	Sniff
L	47474747	Yes	×	×
M	31536000	No	✓	✓
N	31536000	No	✓	✓
O	31536000	No	✓	✓

* เพื่อสงวนชื่อ E-commerce Websites จึงใช้อักษรย่อแทน

จากตารางที่ 5 สำหรับเว็บ E-Commerce จะเห็นว่า 3 ใน 4 เว็บ แม้มีการตั้งค่า HSTS config ด้วยค่า Maxage ที่เหมาะสมเมื่อดูจาก HTTP Response Header แต่ก็โดน

รูปที่ 5 ผลการโจมตี SSL Strip และ Sniff

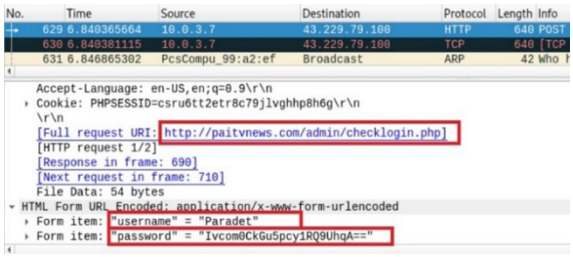
SSL Strip ปลด HTTPS ไปเป็น HTTP ได้ และถูก Sniff รหัสผ่านได้โดยง่าย มีเพียง E-commerce Web L ที่มีการตั้งค่า HSTS Configuration เป็นแบบ Preload ที่ไม่โดน Strip จึงทำให้ไม่สามารถ Sniff รหัสผ่านได้

ตารางที่ 6 ผลการโจมตีเว็บอาสาสมัคร

เว็บไซต์	HSTS		SSL Strip Attack	
	Max-Age	Preload	SSL Strip	Data Sniff
paitvnews.com	31536000	Yes	✓	×
isanmsu.com	ไม่พบ HSTS config.		×	×

จากตารางที่ 6 แสดงผลการโจมตีต่อเว็บอาสาสมัคร 2 เว็บ ได้ผลลัพธ์คือ paitvnews.com ที่เมื่อดูจาก HTTP Response Header เหมือนจะมีการตั้งค่า HSTS แบบ Preload ไว้ ซึ่งน่าจะไม่สามารถ Strip ได้ แต่กลับสามารถ Strip HTTPS ไปเป็น HTTP ได้ แต่ที่น่าแปลกใจคือในทางตรงข้าม คือ isanmsu.com ซึ่งเมื่อดูจาก HTTP Response Header เหมือนไม่มีการตั้งค่า HSTS ไว้เลยกลับไม่สามารถ Strip และ Sniff ได้ ซึ่งจะได้วิเคราะห์ในการตรวจสอบขั้นต่อไป

ที่น่าสังเกตอีกอย่างคือ paitvnews.com แม้โดน Strip ได้ แต่มีการแฮชค่ารหัสผ่านไว้ด้วยเทคนิค Salted-Hash Password (SHP) ทำให้ผลการ Sniff ไม่อาจได้ค่ารหัสผ่านไปใช้ประโยชน์ได้ แสดงดังในรูปที่ 6 ซึ่งเหมือนกับเว็บ Online Banking ของธนาคารแห่งหนึ่งในไทยที่แสดงไว้ในตารางที่ 4



รูปที่ 6 Salted-Hash Password

ซึ่งถือเป็นแนวทางป้องกันที่ดี ในกรณีโพรโทคอล HTTPS หลุด รหัสผ่านก็ยังได้รับการป้องกันชั้นที่ 2

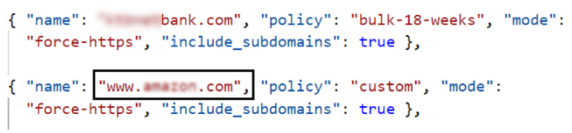
ตารางที่ 7 สรุปผลการโจมตีกลุ่มตัวอย่างทั้ง 17 เว็บ

เว็บไซต์	SSL Strip Attack		
	โจมตีได้	โจมตีไม่ได้	ดักจับไม่ได้
เว็บไซต์ธนาคารในไทย	10	1	2
เว็บไซต์ E-Commerce	3	1	1
เว็บไซต์วิจัยและเว็บไซต์ทีวีออนไลน์	1	1	2

สรุปผลการโจมตีเว็บกลุ่มตัวอย่างที่ใช้ทดลองเป็นไปดังตารางที่ 7 พบว่า SSL Strip Script ใหม่ของ Hacker หรือ Script เดิมของ Marlinspike ก็ให้ผลการโจมตีเหมือนเดิม แต่เป็นที่น่าแปลกใจเพราะเคยมีผลการวิจัย [8], [9] ที่แสดงให้เห็นว่าหลังมีกลไก HSTS เกิดขึ้น SSL Strip เดิมของ Marlinspike ถูกนำมาทดสอบแล้วไม่สามารถเปลือย SSL เปลี่ยน HTTPS ไปเป็น HTTP ได้ และผลของกลุ่มเว็บอาสาสมัคร ยิ่งดูน่าสงสัยเพราะ isanmsu.com ที่ดูจาก HTTP Response Header เหมือนจะไม่มีการคอนฟิก HSTS กลับไม่อาจ Strip ได้ และส่วน paitvnews.com ในทางตรงข้ามที่เหมือนมีการคอนฟิก HSTS แบบ Preload กลับโดน Strip ได้

3.3 ผลการวิเคราะห์ HSTS Preload List

จากประเด็นปัญหาที่พบในตอนท้ายของหัวข้อ 3.2 ผู้วิจัยจึงได้ทำการตรวจสอบข้อมูลเกี่ยวกับการ Preload HSTS และพบว่า เว็บไซต์ที่มีการตั้งค่า HSTS แบบ Preload จะถูกบรรจุไว้ใน List ที่ https://chromium.go oglesource.



รูปที่ 7 List HSTS แบบ Preload ส่วนที่ 1

รูปที่ 8 isanmsu.com อยู่ใน Preload List

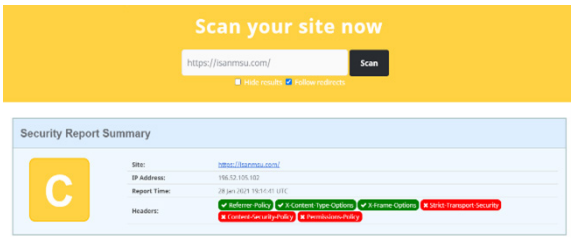
com/chromium/src/net/+/master/http/transport_security_state_static.json และเมื่อเว็บเบราว์เซอร์มีการ Update จะมีการดึงเอา List นี้ไปเก็บไว้ในเบราว์เซอร์

ผลการตรวจสอบ List ดังกล่าวพบว่า เว็บธนาคารออนไลน์ B (ที่มีการตั้งค่าแบบ Preload และไม่ถูก Strip จากการทดลอง) E-commerce Web L (ที่มีการตั้งค่าแบบ Preload และไม่ถูก Strip จากการทดลอง) ล้วนถูกลิสต์ไว้อยู่ใน transport_security_state_static .json ดังแสดงรูปที่ 7 ส่วน paitvnews.com แม้ใน HTTP Response Header เหมือนมีการตั้งค่า HSTS แบบ Preload แต่กลับถูก SSL Strip ได้นั้น ไม่พบชื่อเว็บใน List แต่อย่างใด จึงสรุปได้ว่าแม้มีการตั้งค่า HTTP Header ให้ HSTS เป็น Preload แต่หากไม่ได้ทำการ Preload เข้า List จริง ก็ยังคงถูก SSL Strip ได้

สำหรับ isanmsu.com แม้ไม่พบการตั้งค่า HSTS ใน HTTP Response Header แต่กลับไม่โดน SSL Strip เพราะจากการตรวจสอบ Preload List ดังรูปที่ 8 พบว่า มีการทำ HSTS Preload ไว้

โดยการทดลองก่อนหน้านี้ที่ไม่พบ Header HSTS ของเว็บไซต์ isanmsu.com เพราะหลังลงทะเบียน HSTS Preload เสร็จแล้ว ได้ทำการลบค่า Header HSTS ออก ทำให้สรุปได้ว่า แม้จะลบ HSTS Header หลัง Preload ก็ไม่มีผลในการป้องกัน SSL Strip เพราะ HSTS Header เหมือนจะไม่ได้รับการสนใจเลยด้วยซ้ำจากเว็บเบราว์เซอร์ แต่ดูเหมือนจะอาศัย Preload List แทน

จากการทดลองเว็บที่บริการให้คะแนนความมั่นคงปลอดภัยของเว็บไซต์หลายบริการ เช่น https://security



รูปที่ 9 ผลการ Scan เว็บไซต์ isanmsu.com

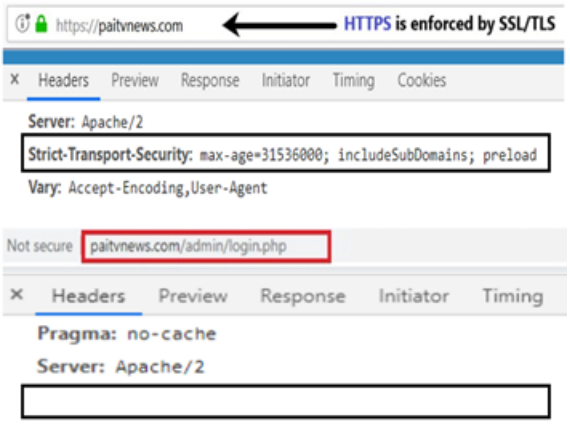
headers.com, www.serpworx.com และ ssllabs.com พบว่า มีข้อผิดพลาดในการวิเคราะห์ HSTS ดังตัวอย่างรูปที่ 9 ที่ให้คะแนน isanmsu.com ว่าไม่ผ่านเรื่อง HSTS เพียงเพราะดูค่าจาก Response Header ทั้งที่เว็บนี้มีความมั่นคงปลอดภัยจาก SSL Stripping Attack ที่ได้ทำ HSTS Preload ไว้และอยู่ใน Preload List เรียบร้อยแล้ว การค้นพบนี้เป็นความรู้สำคัญที่ควรนำไปปรับวิธีการให้คะแนนความมั่นคงปลอดภัยของเว็บไซต์ใหม่

3.4 ผลการวิเคราะห์ Hacker Scripts

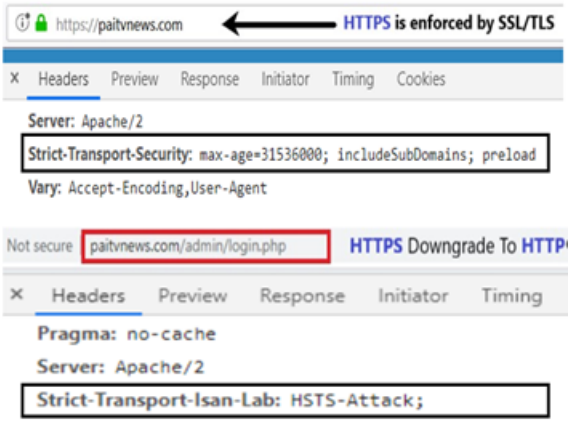
เพื่อให้เข้าใจมากขึ้น จึงได้ทำการตรวจสอบ Scripts ของ Hackers ทั้งแบบเดิมของ Marlinspike และแบบใหม่ของผู้โจมตีที่ใช้ Bettercap จากการศึกษากลไก Script แบบใหม่ พบ Module ที่ชื่อว่า HSTS Hijack ที่ทำให้เข้าใจถึงสาเหตุการล้มเหลวของกลไก HSTS อธิบายดังนี้

HSTS Hijack Module สามารถใช้โจมตีเพื่อลบ HSTS Header ที่ถูกตั้งค่ามาจากเซิร์ฟเวอร์ออกได้ และยังสามารถใช้เพิ่มค่า HSTS Header ตามแต่ Hacker ต้องการหลังแทรกกลางการสื่อสารได้แล้ว

เพื่อให้เข้าใจผลการโจมตีของ Module ดังกล่าว ได้ทำการปรับ Script เพื่อทดลองโจมตี paitvnews.com โดยลบ HSTS Header ที่ตั้งค่ามาจากฝั่งเซิร์ฟเวอร์ออก (ดังแสดงรูปที่ 10) และเพิ่ม HSTS Header เข้าไปโดยที่ไม่มีการตั้งค่ามาก่อนในฝั่งของเซิร์ฟเวอร์ (ดังแสดงรูปที่ 11) สรุปผลได้ว่า Module ดังกล่าวสามารถใช้เพิ่มและลบ HSTS Configuration ได้อย่างง่ายดาย โดยไม่พบว่ามีการแจ้งเตือนความผิดปกติใดๆ จากเว็บเบราว์เซอร์



รูปที่ 10 ผลการ Remove HSTS Header



รูปที่ 11 ผลการ Inject HSTS Header

จากผลการทดลองโดย Module HSTS Hijacking ทำให้เข้าใจได้ชัดเจนว่า Script ของการจู่โจม SSL Strip Attack แบบใหม่ที่มีการเผยแพร่ในกลุ่ม Hacker ในช่วง ค.ศ. 2019 สามารถนำมาใช้ปรับเปลี่ยนค่าคอนฟิกของกลไก HSTS ที่ตั้งค่ามาจากฝั่งเซิร์ฟเวอร์แล้วส่งผ่านเครือข่ายมายังเว็บเบราว์เซอร์ปลายทางได้หมด ทำให้ส่งผลลบได้ 2 ประการ คือ 1) การตั้งค่า HSTS จากเซิร์ฟเวอร์จะไม่มีประโยชน์อะไร เพราะถูก Hacker ลบออกได้ และ 2) Hacker ยังสามารถจู่โจมโดยการเพิ่มค่า HSTS Configuration ตามที่ตัวเองต้องการเข้าไป ซึ่งสามารถนำไปใช้บังคับให้เว็บไซต์ที่ไม่ใช้และไม่ได้สนับสนุน https ถูกบังคับเป็น https และก่อให้

เกิดการล้มเหลวในการเข้าสู่เว็บไซต์นั้นได้ หรือก็คือ Denial of Service (DoS) Attack เว็บไซต์นั้นได้

ด้วยเหตุนี้ จะเห็นได้จากผลการทดลองก่อนหน้านี้ เว็บเบราว์เซอร์ต่างๆ ได้ทำการยกเลิกการสนับสนุนการทำงานของ HSTS Configuration ที่อยู่ใน HTTP Header ที่ตั้งค่าจากเซิร์ฟเวอร์แล้วส่งผ่านเครือข่ายมายังเว็บเบราว์เซอร์ปลายทางทั้งหมด ดังนั้นแม้มีการตั้งค่าใดๆ ก็ไม่มีผล และจากผลการทดลองและตรวจสอบ Response Header จะเห็นว่ามีเว็บไซต์จำนวนมากยังไม่ทราบปัญหานี้ ซึ่งยังอาศัยการตั้งค่า HSTS ให้ทำงานในฝั่งเซิร์ฟเวอร์ที่ไม่ได้ผลในการป้องกัน

แนวทางที่ถูกต้องในปัจจุบัน คือ หากเว็บใดต้องการบังคับใช้ HSTS เพื่อต่อต้าน SSL Stripping Attack จะต้องทำการ Preload โดยตั้งค่า HSTS Header เป็น Preload แล้วทำการลงทะเบียนกับ https://hstspreload.org เมื่อดำเนินการเรียบร้อยแล้วต้องรอจนเว็บเบราว์เซอร์ Update Version ถึงจะทำการดึงฐานข้อมูล HSTS Preload List ลงมาและสนับสนุนการบังคับใช้ https ของเว็บที่ทำการ Preload ทั้งนี้ HSTS Header ที่ตั้งค่าไว้หลังจากนั้น จะลบออกก็ได้ ไม่ได้มีผลต่อการบังคับใช้ https แต่อย่างใด

4. สรุป

จากการทดลองวิเคราะห์ปัญหาการทำงานที่ผิดปกติของกลไก HSTS และการกลับมาโจมตีได้ใหม่ของ SSL Stripping Attack ในการทดลองสามารถสรุปผลได้ดังนี้

1) เว็บไซต์ธนาคารออนไลน์ และเว็บไซต์ E-Commerce หลายแห่งยังใช้งานกลไก HSTS ที่เป็นเทคโนโลยีในการใช้ป้องกัน SSL Stripping Attack ยังไม่ถูกต้อง เนื่องจากตั้งค่า HSTS ที่เว็บเซิร์ฟเวอร์แบบเดิมที่ไม่ได้รับการสนับสนุนแล้ว และบางเว็บไซต์แย่ถึงขั้นไม่มีการ Configuration กลไก HSTS เลย มีเพียง 1 เว็บธนาคารออนไลน์ในประเทศไทยจาก 11 แห่ง และ 1 เว็บ E-commerce จาก 4 แห่ง ที่ทดสอบพบมีการ Preload HSTS อย่างถูกต้อง

2) การจัดการ HSTS ที่เหมาะสมเพื่อช่วยป้องกัน SSL Stripping Attack ต้องทำการ Preload เว็บไซต์ว่าใช้ Strict HTTPS ที่ hstspreload.org ซึ่งการตั้งค่า HSTS Configuration ที่เว็บเซิร์ฟเวอร์ไม่มีประโยชน์อีกต่อไป เพราะฝ่าย Hacker สามารถใช้เทคนิค HSTS Hijacking ในการปลดค่าออกได้

3) เว็บเบราว์เซอร์ในปัจจุบัน ไม่สนับสนุนการตั้งค่า HSTS จาก HTTP Response Header ที่ส่งผ่านอินเทอร์เน็ตอีกแล้ว เนื่องจากเทคนิค HSTS Hijacking สามารถก่อปัญหา DoS ต่อเว็บที่ไม่ใช่ HTTPS ได้ ด้วยเหตุนี้เอง จึงทำให้ SSL Stripping Attack Script เดิมของ Marlinspike ที่ใช้ไม่ได้ผล กลับมาใช้ได้ผลอีกครั้ง

4) ระบบ Scan Website ที่ใช้ตรวจสอบการป้องกัน SSL Stripping Attack ว่าป้องกันได้หรือไม่ ใช้วิธีเช็คแค่ HTTP Header HSTS ไม่ได้ผลอีกอีกต่อไป ต้องทำการตรวจสอบในฐานข้อมูล HSTS Preload List จึงจะได้ผลที่ถูกต้อง

5) จากการทดลอง พบว่ามีบางเว็บไซต์ที่ทำการปรับใช้กลไกป้องกันชั้นที่ 2 คือ Salted-hash password (SHP) ซึ่งเป็นกลไกที่ดีช่วยทำให้ข้อมูลที่ Hacker ดักจับ เช่น รหัสผ่าน ไม่อยู่ในรูปแบบ Clear Text แม้ Hacker จะ Strip HTTPS เป็น HTTP ได้ ซึ่งวิธีนี้น่าจะเป็นแนวทางที่ควรทำควบคู่ไปกับการ Preload HSTS

ในส่วนของ SHP แม้จะเป็นแนวทางที่ดี แต่ยังมีโอกาสโดนโจมตีด้วย Rainbowcrack [17] ได้ งานวิจัยที่จะทำต่อไปในอนาคต จะได้มีการเสนอวิธีการที่เหมาะสมในการแก้ปัญหาต่อไป

5. กิตติกรรมประกาศ

ขอขอบพระคุณ กรมสอบสวนคดีพิเศษ (DSI) กระทรวงยุติธรรม ในการร่วมวิจัย งานวิจัยนี้ได้รับทุนสนับสนุนส่วนหนึ่งจากทุนวิจัยรายได้ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

เอกสารอ้างอิง

[1] E. Rescorla. (2000). *HTTP Over TLS*. [Online]. Available: https://www.rfc-editor.org/info/rfc2818/

[2] E. Rescorla. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. [Online]. Available:

https://www.rfc-editor.org/info/- rfc8446/

[3] C. Adams and S. Lloyd, *Understanding PKI: Concepts Standards and Deployment Con Siderations*, Addison Wesley, 2002, pp. 11–15.

[4] M. Marlingspike. (2009, August). New Tricks for Defeating SSL in Practice. Black Hat, USA. [online]. Available: https://www.blackhat.com /presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

[5] A. Fung and K. Cheung, "SSLock: Sustaining the trust on entities brought by SSL," in *Proceedings of the ACM Symposium on Information*, 2010, pp. 204–213.

[6] N. Nikiforakis, Y. Younan, and W. Joosen, "HProxy: Client-side detection of SSL stripping attack," in *Proceedings of International Conference on Detection of Intrusions*, 2010, pp. 200–218.

[7] A. P. H. Fung and K. W. Cheung, "HTTPSLock: Enforcing HTTPS in unmodified browsers with cached Javascript," in *Proceedings of Fourth International Conference on Network and System Security*, 2010, pp. 269–274.

[8] S. Puangpronpitag and N. Sriwiboon, "Simple and lightweight HTTPS enforcement to protect against SSL striping attack," in *Proceedings of International Conference on Computational Intelligence, Communication Systems and Networks*, Phuket, Thailand, 2012, pp. 229–234.

[9] S. Puangpronpitag and A. Tooltham "Experimental evaluation of SSL stripping attack solutions," *Information Technology Journal*, vol. 10, no. 1, pp. 37–47, 2014 (in Thai).

[10] A. Tooltham and S. Puangpronpitag, "Click2-Enforce: A browser extension to protect against SSL stripping attacks," *Information Technology Journal*, vol. 9, no. 2, pp. 7–13, 2013 (in Thai).

[11] J. Hodges, C. Jackson, and A. Barth. (2012). *HTTP Strict Transport Security (HSTS)*. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6797/

[12] S. Puangpronpitag, "Surveys of e-banking web security," Information Security & Advanced Network Research Group, Tech. Rep. 2019-1005, Oct. 2019.

[13] W. Jacqueem. (2021, January). Force HSTS Using htaccess. InMotion Hosting, Virginia Beach. [Online]. Available: https://www.in-motionhosting.com/support/website/force-hsts-using-htaccess/

[14] T. Griffin. (2020, December). How to Enable HTTP Strict Transport Security (HSTS) in WordPress, Griffin Media LLC, United States. [Online]. Available: https://thomasgriffin.com/enable-http-strict-transport-security-hsts-wordpress/

[15] Bettercap. (2019). bettercap Version 2.26.1. [Online]. Available: https://bettercap.org

[16] Google Inc. (2020). *Chromium HSTS*. [Online]. Available: https://hstspreload.org/

[17] RainbowCrack Project. (2020). *RainbowCrack*. [Online]. Available: http://project-rainbowcrack .com/