กลไกการตรวจจับและป้องกันการจู่โจมเว็บแอพพลิเคชันรูปแบบใหม่ โดยการบังคับใช้เอชทีทีพีเอส

A New Detection and Protection Mechanism Against Web Application Attack with HTTPS Enforcement

สมนึก พ่วงพรพิทักษ์[1]* และ ณัฐวุฒิ ศรีวิบูลย์[2]
Somnuk Puangpronpitag[1]* and Nattavut Sriwiboon[2]

บทคัดย่อ

HypertextTransferProtocol (HTTP) overTransport Layer Security (TLS) หรือ HTTPS เป็นวิธีการสื่อสารข้อมูลที่มั่นคงเพื่อป้องกันเว็บแอพพลิเคชันจากการโจมตีดักจับข้อมูล การจู่โจมแบบเล่นซ้ำคุ้กกี้ และการจู่โจมแบบ Sidejacking อย่างไรก็ตามการโจมตีด้วยวิธี TLS Strip เป็นวิธีการที่ผู้โจมตีนำมาใช้เพื่อโจมตี HTTPS จากการศึกษางานวิจัยก่อนหน้านี้ มีการเสนอวิธีการเพื่อแก้ไขปัญหาการโจมตีด้วยวิธี TLS Strip แต่ยังมีข้อบกพร่องด้านประสิทธิภาพในการป้องกัน ในงานวิจัยนี้จึงเสนอเกณฑ์ในการประเมินผลการแก้ไขปัญหาของงานวิจัยก่อนหน้านี้และเสนอวิธีการเพื่อแก้ไขปัญหา โดยมีการทดสอบประสิทธิภาพของวิธีการที่เสนอในงานวิจัยนี้ ผลการทดสอบแสดงถึงประสิทธิภาพของวิธีการที่เสนอเพื่อแก้ไขปัญหาการโจมตีเว็บไซต์ด้วยวิธี TLS Strip

คำสำคัญ: HTTPS การจู่โจมแบบเล่นซ้ำคุ้กกี้ Sidejacking, TLS Strip

Abstract

Hypertext Transfer Protocol (HTTP) over Transport Layer Security (TLS) or HTTPS is a secure communication method to protect web application against eavesdropping attacks, such as cookie replay attacks, sidejacking attacks, sniffing attacks. However, the TLS stripping technique has been deployed by attackers to bypass the HTTPS. According to the literature, there have been several solutions recently proposed to fix the TLS stripping problem; however all of them still have some critical weaknesses. Consequently, we have proposed in this paper an evaluation of the previous solutions. After that, a new solution has been proposed to fix the weakness. The evaluation of our new solution on a test-bed network has also been carried out. The experimental results have revealed favorable features regarding our solution.

[1] อาจารย์ Information Security & Advanced Network (ISAN) Group คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

[2] ผู้ช่วยวิจัย Information Security & Advanced Network (ISAN) Group คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

* Corresponding Author, Tel. 08-9453-2159, E-mail: somnuk.p@msu.ac.th

รับเมื่อ 24 เมษายน 2555 ตอบรับเมื่อ 30 กรกฎาคม 2555

Keyword: HTTPS, Cookie Replay Attack, Sidejacking, TLS stripping

1. บทนำ

Hyper Text Transfer Protocol (HTTP) [1] เป็นรูปแบบการสื่อสารข้อมูลบนเทคโนโลยีเว็บไซต์ โดยระบบมีรูปแบบการทำงานใช้เว็บเซิร์ฟเวอร์ในการกระจายข้อมูลไปที่เครื่องไคลเอนต์ ซึ่งในการร้องขอข้อมูลจากเครื่องไคลเอนต์จะใช้เว็บเบราว์เซอร์ในการกำหนดชื่อเว็บไซต์ แล้วส่งการร้องขอไปที่เว็บเซิร์ฟเวอร์ เมื่อคำขอมาถึงเว็บเซิร์ฟเวอร์ก็จะประมวลผลโดยการค้นหาไฟล์ที่ระบุในชื่อเว็บไซต์แล้วตอบกลับไปที่ไคลเอนต์ เมื่อข้อมูลตอบกลับมาถึงเครื่องไคลเอนต์ก็จะนำข้อมูลแสดงผลในเว็บเบราว์เซอร์ ซึ่งข้อมูลที่ส่งผ่านระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์จะอยู่ในรูปของ Clear Text เมื่อเว็บไซต์ถูกโจมตีด้วยวิธีแทรกกลางการสื่อสาร (Man In The Middle or Monkey In The Middle: MITM) [2] ผลของการโจมตีคือผู้โจมตีสามารถดักจับข้อมูลที่สื่อสารบน HTTP ซึ่งอาจเป็นข้อมูลสำคัญของผู้ใช้เช่นชื่อบัญชีผู้ใช้ รหัสผ่าน ค่าคุ้กกี้หรือหมายเลขระบุการสื่อสาร (Session ID) โดยการโจมตีเพื่อดักจับข้อมูลเป็นพื้นฐานที่นำมาสู่การโจมตีเว็บไซต์หลายรูปแบบเช่น การจู่โจมแบบเล่นซ้ำคุ้กกี้ (Cookie Replay Attack) และวิธีการจู่โจมแบบ Sidejacking [3]

HTTPS [4] เป็นรูปแบบการสื่อสารข้อมูลบนระบบเว็บไซต์ ที่เสนอเพื่อแก้ไขปัญหาการทำงานของ HTTP ที่ไม่มีความมั่นคงในการใช้งาน โดยการนำ Transport Layer Security (TLS) [5] เป็นโพรโทคอลที่ใช้ในการเข้ารหัสข้อมูลการสื่อสารระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์ ทำให้การสื่อสารมีความมั่นคงในการใช้งาน โดยรูปแบบการทำงานของ HTTPS มีการสร้างช่องทางการสื่อสารที่ปลอดภัย โดยการติดตั้งกุญแจสาธารณะของ Certificate Authority (CA) ไว้บนเว็บเบราว์เซอร์เพื่อใช้พิสูจน์ใบรับรอง (Certificate) ของเว็บไซต์ และก่อนการส่งข้อมูลระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์ จะนำข้อมูลต้นฉบับมาเข้ารหัสเพื่อเปลี่ยนข้อมูลให้อยู่ในรูป Cipher Text ทำให้การทำงานของระบบเว็บไซต์สามารถพิสูจน์ตัวตน (Authentication) ระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์ การรักษาข้อมูลให้คงสภาพเพื่อยืนยันความถูกต้องของข้อมูล (Integrity) การป้องกันข้อมูลให้เป็นความลับ (Confidentiality) เพื่อป้องกันปัญหาการโจมตีดักจับและเปลี่ยนแปลงข้อมูล

อย่างไรก็ตาม การใช้งานเว็บไซต์บนโพรโทคอล HTTPS สามารถถูกโจมตีด้วยเทคนิคและวิธีการต่างๆ ที่มีพื้นฐานมาจากการโจมตีแบบ MITM โดย SSL Sniff เป็นวิธีการโจมตี HTTPS ที่ใช้หลักการส่ง Certificate ปลอมที่ถูกสร้างขึ้นโดยผู้โจมตีไปที่เครื่องเหยื่อ ซึ่งผลของการโจมตีบนเว็บเบราว์เซอร์ของเหยื่อจะแสดงข้อความเพื่อเตือนผู้ใช้ว่าเว็บไซต์ใช้งาน Certificate ที่ไม่ปลอดภัย ผู้ใช้สามารถปฏิเสธการใช้งานหรือดำเนินการตามคำเตือนบนเว็บเบราว์เซอร์ แต่ในปี ค.ศ. 2009 Marlingspike [6] ได้เสนอ SSL Strip หรือ TLS Strip เป็นวิธีที่ใช้โจมตีเว็บไซต์ที่ทำงานบน HTTPS โดยผลการโจมตีบนเบราว์เซอร์ของเหยื่อไม่แสดงข้อความแจ้งเตือนเนื่องจากการทำงานบนเว็บเบราว์เซอร์ดำเนินบน HTTP ทำให้ผู้ใช้ไม่สามารถตรวจสอบการทำงานของระบบเว็บไซต์ว่าดำเนินบน HTTPS ที่ปลอดภัยหรือไม่

จากปัญหาการโจมตีด้วยวิธี SSL Strip ซึ่งเป็นปัญหาการโจมตี HTTPS ที่ส่งผลเสียต่อระบบเว็บไซต์ เนื่องจากเบราว์เซอร์ไม่สามารถตรวจสอบและแจ้งเตือนการโจมตี มีงานวิจัยก่อนหน้านี้ที่เสนอวิธีการเพื่อป้องกันการโจมตี แต่ประสบปัญหาเช่นความสะดวกในการใช้งาน ความสามารถในการปรับใช้กับระบบปัจจุบัน ปัญหาระบบไม่รองรับการทำงานกับทุกเว็บเบราว์เซอร์ และประสิทธิภาพในการป้องกัน

ดังนั้นในงานวิจัยนี้จึงเสนอระบบป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จากการโจมตีแบบ MITM และการโจมตีด้วยวิธี SSL Strip โดยมีการประเมินวิธีการของงานวิจัยก่อนหน้านี้และเสนอวิธีการป้องกัน โดยการออกแบบให้ระบบสามารถตรวจสอบ และบังคับใช้โพรโทคอล HTTPS ในการสื่อสารด้วยเว็บเซิร์ฟเวอร์ ทำให้ระบบที่เสนอมีความสะดวกในการใช้งาน รองรับ

การทำงานกับทุกเว็บเบราว์เซอร์ และมีประสิทธิภาพในการป้องกันการโจมตี โดยมีการพัฒนาระบบต้นแบบและทดสอบระบบที่เสนอในงานวิจัยนี้เพื่อแสดงให้เห็นถึงประสิทธิภาพในการใช้งาน ประสิทธิภาพในการป้องกันการโจมตีเว็บไซต์จากการโจมตีแบบ MITM และการโจมตีด้วยวิธี SSL Strip

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ความมั่นคงในการใช้งานเว็บไซต์

จากการโจมตีระบบเว็บไซต์ด้วยวิธี MITM ที่สามารถดักจับข้อมูลในการสื่อสาร เช่นข้อมูลชื่อบัญชีผู้ใช้ และรหัสผ่านในขั้นตอนที่เหยื่อเข้าใช้งานระบบบนโพรโทคอล HTTP แต่ในการทำงานของระบบเว็บไซต์ยังมีข้อมูลสำคัญอื่นๆ ที่ใช้ในการตรวจสอบสิทธิ์การเข้าใช้งานระบบโดยหลังจากที่ไคลเอนต์ตรวจสอบสิทธิ์ในการเข้าใช้งานระบบด้วยการส่งชื่อบัญชีผู้ใช้ และรหัสผ่านมาที่เว็บเซิร์ฟเวอร์แล้ว การทำงานของเว็บเซิร์ฟเวอร์จะสร้างคุ้กกี้เพื่อใช้เก็บค่าเช่นชื่อบัญชีผู้ใช้ รหัสผ่าน และหมายเลขระบุการสื่อสาร (Session ID) แล้วส่งข้อมูลคุ้กกี้ไปจัดเก็บบนเว็บเบราว์เซอร์ เมื่อไหร่ก็ตามที่เว็บเบราว์เซอร์ร้องขอข้อมูลก็จะส่งค่าคุ้กกี้เพื่อแสดงให้เว็บเซิร์ฟเวอร์เห็นว่าไคลเอนต์ผ่านขั้นตอนการตรวจสอบสิทธิ์ในการเข้าใช้งานระบบเว็บไซต์อย่างถูกต้อง ซึ่งการส่งข้อมูลคุ้กกี้ระหว่างเว็บเบราว์เซอร์กับเว็บเซิร์ฟเวอร์ที่ระบบทำงานบน HTTP ข้อมูลจะอยู่ในรูปของ Clear Text ด้วยเหตุนี้เมื่อผู้โจมตีใช้วิธี MITM เพื่อดักจับข้อมูลคุ้กกี้แล้วใช้วิธีการจู่โจมแบบเล่นซ้ำคุ้กกี้ และวิธีการจู่โจมแบบ Sidejacking เพื่อดักจับข้อมูลคุ้กกี้รวมถึงหมายเลขระบุการสื่อสารแล้ว ส่งซ้ำไปยังเว็บเซิร์ฟเวอร์ผู้โจมตีสามารถเข้าใช้งานระบบเว็บไซต์โดยถือสิทธิ์เป็นเหยื่อและสามารถจัดการข้อมูลในระบบเว็บไซต์ได้เทียบเท่ากับเหยื่อทุกประการ

จากปัญหาการจู่โจมเว็บไซต์ด้วยวิธีการจู่โจมแบบเล่นซ้ำคุ้กกี้และการจู่โจมแบบ Sidejacking มีงานวิจัยก่อนหน้านี้ที่เสนอเพื่อแก้ไขปัญหาแต่ในงานวิจัย [7] ได้เสนอเกณฑ์ และวิธีการทดสอบที่เหมาะสมเพื่อทดสอบและวิเคราะห์ปัญหาของระบบในงานวิจัยที่เสนอเพื่อแก้ไขปัญหาการจู่โจมแบบเล่นซ้ำคุ้กกี้ และการจู่โจมด้วยวิธี Sidejacking พบว่าวิธีการต่างๆ ที่เสนอไม่มีประสิทธิภาพในการป้องกัน

2.2 HTTP over TLS

HTTP over TLS หรือ HTTPS เสนอแนวคิดการสร้างช่องทางการสื่อสารข้อมูลบนเว็บไซต์ที่ปลอดภัย โดยปกติแล้วเว็บไซต์ที่ทำงานบน HTTP ข้อมูลที่ถูกส่งผ่านระหว่างการสื่อสารอยู่ในรูปของ Clear Text โดยการเรียกใช้งานเว็บไซต์บน HTTP ผ่านเว็บเบราว์เซอร์จะระบุที่อยู่ของเว็บไซต์เช่น http://www.test.com ส่วนการเรียกใช้งานเว็บไซต์ที่ทำงานบน HTTPS เว็บเบราว์เซอร์จะแสดงที่อยู่ของเว็บไซต์ เช่น https://www.test.com ซึ่งข้อมูลที่สื่อสารบน HTTPS จะถูกเข้ารหัสเพื่อความปลอดภัยในการใช้งานเว็บไซต์ โดยอาศัยหลักการเข้ารหัสร่วมกันระหว่างการเข้ารหัสแบบอสมมาตรและการเข้ารหัสแบบสมมาตร

สำหรับบางเว็บไซต์มีการนำ Extended Validation (EV) SSL certificate หรือ EV-SSL [8] ซึ่งเป็นมาตรฐานของ x.509 digital certificates เพื่อเพิ่มกลไกการป้องกันการโจมตี HTTPS โดย EV-SSL เป็นแถบที่แสดงข้อมูลของ Certificate ที่ใช้งานในแต่ละเว็บไซต์ เช่นข้อมูลชื่อผู้ให้บริการเว็บไซต์ โดยจะแสดงแถบสีเขียวบนเว็บเบราว์เซอร์สำหรับเว็บไซต์ที่สื่อสารข้อมูลบน HTTPS ด้วย Valid Certificate ทำให้เว็บเบราว์เซอร์สามารถขยายการใช้งาน HTTPS เพื่อโต้ตอบกับผู้ใช้

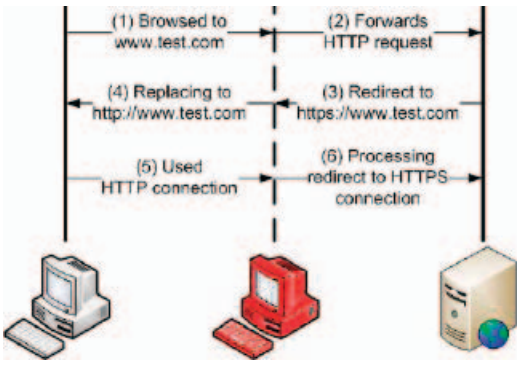
ในขั้นตอนการสื่อสารข้อมูลของเว็บไซต์ที่ทำงานบน HTTPS ก่อนการเริ่มต้นสื่อสารข้อมูลระหว่างไคลเอนต์กับเว็บเซิร์ฟเวอร์ ในขั้นตอนแรกเมื่อผู้ใช้ต้องการเรียกใช้งานเว็บไซต์ โดยปกติจะใช้เว็บเบราว์เซอร์โดยมีการระบุชื่อเว็บไซต์เช่น test.com หรือ www.test.com แล้วเว็บเบราว์เซอร์จะประมวลผลเริ่มต้นการสื่อสารโดยส่งการร้องขอไปที่เว็บเซิร์ฟเวอร์บน HTTP และเมื่อเว็บเซิร์ฟเวอร์ได้รับคำขอก็จะประมวลผลการร้องขอแล้วตอบกลับข้อมูลบนโพรโทคอล HTTPS พร้อมกับ

ส่ง Certificate ของเว็บไซต์มาที่เว็บเบราว์เซอร์ แล้วระบบจึงเริ่มขั้นตอน SSL Handshake ในการสื่อสาร โดยเว็บเบราว์เซอร์จะประมวลผลเพื่อพิสูจน์ Certificate ของเว็บไซต์โดยการใช้กุญแจสาธารณะของ CA ที่ติดตั้งบนเว็บเบราว์เซอร์ และขั้นตอนการตกลงกุญแจสำหรับใช้ในกระบวนการเข้ารหัส ถอดรหัสข้อมูลที่ปลอดภัย

2.3 การจู่โจม HTTPS

จากรูปแบบการทำงานของเว็บไซต์บน HTTPS จะเห็นได้ว่าเมื่อมีการส่งคำขอไปที่เว็บเซิร์ฟเวอร์ การทำงานของเว็บเบราว์เซอร์เริ่มต้นสื่อสารบนโพรโทคอล HTTP เมื่อคำขอไปถึงเว็บเซิร์ฟเวอร์ ระบบจึงดำเนินการสื่อสารบน HTTPS ดังนั้นระบบเว็บไซต์ที่ทำงานบน HTTPS มีการกำหนดการสื่อสารบนโพรโทคอล HTTPS และตอบกลับ Certificate ที่ตำแหน่งเว็บเซิร์ฟเวอร์เท่านั้น ด้วยเหตุนี้เมื่อผู้โจมตีใช้วิธีโจมตีแบบ MITM แล้วโจมตีเว็บไซต์ด้วยวิธี SSL Sniff ทำให้ผู้โจมตีสามารถดักจับข้อมูลสำคัญเช่นชื่อบัญชีผู้ใช้ และรหัสผ่านของเหยื่อได้ เนื่องจากหลักการโจมตีด้วยวิธี SSL Sniff ที่ผู้โจมตีสามารถสร้าง Certificate ปลอมเพื่อส่งไปที่เครื่องเหยื่อในขั้นตอน SSL Handshake และเหยื่อใช้ Certificate ของผู้โจมตีในกระบวนการเข้ารหัสข้อมูล เมื่อเหยื่อส่งข้อมูลไปที่เว็บเซิร์ฟเวอร์ ข้อมูลก็จะผ่านเครื่องผู้โจมตีก่อน ผู้โจมตีก็สามารถถอดรหัสได้ด้วยกุญแจลับที่ถูกจัดเก็บบนเครื่องผู้โจมตี โดยผลของการโจมตีบนเว็บเบราว์เซอร์ของเหยื่อจะแสดงข้อมูลแจ้งเตือนการใช้งานใบรับรองของเว็บไซต์ไม่ปลอดภัย

อย่างไรก็ตาม เว็บไซต์ที่ทำงานบน HTTPS เมื่อถูกโจมตีด้วยวิธี SSL Sniff สามารถป้องกันการโจมตีได้ โดยกระบวนการทำงานของเว็บเบราว์เซอร์ที่สามารถตรวจสอบการโจมตีโดยแสดงข้อมูลเว็บไซต์ใช้งานใบรับรองที่ไม่ปลอดภัย และแสดงคำแนะนำให้ผู้ใช้ปฏิบัติตามขั้นตอนการใช้งานเว็บไซต์ให้มีความปลอดภัย โดยการปฏิเสธการใช้งานเว็บไซต์



รูปที่ 1 หลักการโจมตีด้วยวิธี SSL Strip

2.4 การจู่โจม HTTPS ด้วยวิธี SSL Strip

สำหรับวิธี SSL Strip ที่เสนอเพื่อโจมตีเว็บไซต์ที่ทำงานบน HTTPS โดยจากหลักการทำงานของ HTTPS ที่มีการกำหนดให้ระบบเว็บไซต์ทำงานบนโพรโทคอล HTTPS ที่ตำแหน่งเว็บเซิร์ฟเวอร์เท่านั้น ด้วยเหตุนี้ SSL Strip จึงอาศัยจุดอ่อนที่เว็บเบราว์เซอร์ไม่สามารถตรวจสอบและกำหนดรูปแบบการสื่อสารบนโพรโทคอล HTTPS กับเว็บเซิร์ฟเวอร์ โดยหลักการโจมตีด้วยวิธี SSL Strip เพื่อดักจับข้อมูลสำคัญของเหยื่อที่ใช้ในการสื่อสารกับเว็บเซิร์ฟเวอร์ แสดงรายละเอียดดังรูปที่ 1

จากรูปที่ 1 เป็นรูปแบบการโจมตีด้วยวิธี MITM และการใช้ SSL Strip ในการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จะเห็นได้ว่าเมื่อเหยื่อถูกโจมตีเว็บเบราว์เซอร์ใช้โพรโทคอล HTTP ในการสื่อสารโดยข้อมูลต่างๆ ที่เหยื่อส่งไปที่เว็บเซิร์ฟเวอร์จะผ่านเครื่องผู้โจมตีก่อน ซึ่งผู้โจมตีสามารถดักจับข้อมูลสำคัญของเหยื่อเช่นชื่อบัญชีผู้ใช้และรหัสผ่านได้ เนื่องจากข้อมูลที่สื่อสารบน HTTP อยู่ในรูปของ Clear Text จากนั้นการทำงานของ SSL Strip จะนำข้อมูลของเหยื่อเข้ารหัสด้วยโพรโทคอล HTTPS แล้วส่งต่อไปที่เว็บเซิร์ฟเวอร์ ด้วยเหตุนี้จึงทำให้เว็บเซิร์ฟเวอร์ไม่สามารถตรวจสอบได้ว่าการสื่อสารที่เกิดขึ้นกับเหยื่อดำเนินการสื่อสารบน HTTP หรือ HTTPS โดยผลของการโจมตีที่เว็บเบราว์เซอร์ของเหยื่อมีความแตกต่างจากวิธี SSL Sniff เนื่องจากเมื่อเว็บไซต์ถูกโจมตีด้วย SSL Strip เบราว์เซอร์ไม่สามารถตรวจสอบและแสดงข้อความ

รูปที่ 2 เปรียบเทียบเว็บไซต์ปกติที่แสดง HTTPS และใช้ EV-SSL เพื่อแสดงข้อมูลใบรับรองกับผลของเว็บไซต์เมื่อถูกโจมตีด้วย SSL Strip

แจ้งเตือน รวมถึงเหยื่อสามารถสื่อสารกับเว็บเซิร์ฟเวอร์ได้ตามปกติบนโพรโทคอล HTTP โดย Cheng และคณะ [9] ได้ทดสอบโจมตีด้วยวิธี SSL Strip ผลการโจมตีสามารถดักจับชื่อบัญชีผู้ใช้ และรหัสผ่านของเหยื่อได้ 100% ดังนั้นจะเห็นได้ว่า SSL Strip เป็นปัญหาที่สำคัญต่อความมั่นคงของระบบเว็บไซต์ที่ทำงานบนโพรโทคอล HTTPS ดังรูปที่ 2 เป็นการแสดงผลของหน้าเว็บไซต์บนเว็บเบราว์เซอร์ของเหยื่อเปรียบเทียบกรณีการใช้งานเว็บไซต์ปกติและเมื่อถูกโจมตีด้วยวิธี SSL Strip

2.5 งานวิจัยที่เกี่ยวข้อง

Nikiforakis และคณะ [10] ได้เสนอ HProxy ที่พัฒนาระบบด้วย Client-side Proxy เพื่อป้องกันการโจมตีด้วยวิธี MITM และการโจมตีด้วย SSL Strip โดยมีการพัฒนาและติดตั้งระบบเพื่อจัดเก็บประวัติการเข้าใช้งานเว็บไซต์เพื่อตรวจสอบว่า เว็บไซต์ที่ผู้ใช้กำลังใช้งานทำงานบน HTTPS หรือไม่หากตรวจสอบพบว่าเว็บไซต์ที่ทำงานบน HTTPS ที่จัดเก็บใน HProxy ไม่ตรงกับที่ผู้ใช้กำลังใช้งานแล้วระบบจะแสดงผลที่เว็บเบราว์เซอร์ว่าเป็นการโจมตีด้วยวิธี MITM อย่างไรก็ตามในการใช้งาน HProxy จะต้องติดตั้งเครื่องมือ client-side proxy บนเครื่องไคลเอนต์ มีการกำหนดการทำงานเพิ่มเติมด้วยภาษา Javascript บนเว็บเซิร์ฟเวอร์เพื่อใช้ในกระบวนการตรวจสอบการโจมตี ระบบสามารถตรวจสอบการโจมตีได้เท่านั้นและผลการทดสอบพบว่าการทำงานของระบบไม่มีประสิทธิภาพในการป้องกัน

Fung และคณะ [11] ได้เสนอ SSLock โดยเป็นข้อเสนอวิธีการเพื่อบังคับใช้ SSL กับเว็บไซต์โดยการพิจารณาการแบ่ง Domain Name สำหรับการใช้งาน SSL โดยผู้พัฒนาเว็บไซต์จะต้องกำหนดรูปแบบการทำงานของระบบเว็บไซต์ตามกระบวนการที่ SSLock เสนอเพื่อตอบกลับ HTTP Header ชื่อ SSLock-Candidates ที่จัดเก็บค่า Domain Name เช่น gmail.com พร้อมด้วย Javascript ที่ใช้อ่านค่า HTTP Header แล้วการทำงานของเบราว์เซอร์จะเปลี่ยนชื่อ Domain Name และส่งคำขอไปที่เว็บเซิร์ฟเวอร์ด้วย Domain Name ที่ปลอดภัยบน HTTPS เช่น https://secure.gmail.com อย่างไรก็ตาม สำหรับวิธีของ SSLock ยังมีข้อเสียในการพัฒนาระบบให้เป็นมาตรฐาน และสามารถป้องกันการโจมตีด้วยวิธี SSL Strip เฉพาะ Domain Name ที่ถูกกำหนดใน SSLock-Candidates เท่านั้น

Fung และคณะ [12] ได้เสนอ HTTPSLock เป็นข้อเสนอวิธีการบังคับใช้งานโพรโทคอล HTTPS สำหรับเว็บไซต์ที่ใช้ Valid Certificate โดยใช้ Javascript ในการตรวจสอบการใช้งาน Certificate หากเว็บไซต์มีการใช้งาน Invalid Certificate ซึ่งอาจเกิดจากการโจมตีแบบ SSL Sniff การทำงานของ HTTPSLock จะไม่อนุญาตให้มีการใช้งานเว็บไซต์ และสำหรับเว็บไซต์ที่ใช้ Valid Certificate แต่เมื่อเข้าใช้งานเว็บไซต์โดยที่มีการแสดงผลของโพรโทคอลบนเว็บเบราว์เซอร์เป็น HTTP การทำงานของ HTTPSLock จะแสดงข้อความบนเว็บเบราว์เซอร์ให้ผู้ใช้ได้ทราบว่ากำลังใช้งานเว็บไซต์ที่ไม่ปลอดภัย อย่างไรก็ตามสำหรับวิธีการของ HTTPSLock ยังมีข้อเสียคือสามารถตรวจสอบการโจมตี HTTPS ได้เท่านั้น โดยเฉพาะการโจมตีด้วยวิธี SSL Strip ปัญหาการใช้งานเนื่องจากระบบรองรับการใช้งาน 70% ของเว็บเบราว์เซอร์ทั้งหมด และเมื่อถูกโจมตีผู้ใช้ไม่สามารถเข้าใช้งานเว็บไซต์ได้ตามปกติเนื่องจากการทำงานของระบบ HTTPSLock จะเปลี่ยนแปลงจากหน้าเว็บไซต์ปกติเป็นหน้าเว็บไซต์ที่แสดงข้อความแจ้งเตือนการโจมตี

Hodges และคณะ [13] เสนอระบบ HTTP Strict Transport Security (HSTS) ในปี ค.ศ. 2012 เป็นวิธีการที่พัฒนามาจากวิธี ForceHTTPS [14] เพื่อแก้ไขปัญหาการโจมตี HTTPS เช่นการโจมตีด้วยวิธี SSL Strip โดยมี

การกำหนดรูปแบบการทำงานให้เว็บเบราว์เซอร์สามารถตรวจสอบและบังคับการใช้งานเว็บไซต์บน HTTPS โดยการพัฒนาระบบที่เซิร์ฟเวอร์มีการกำหนดการตอบกลับ HTTP Header และการทำงานของเว็บเบราว์เซอร์จะประมวลผลเพื่อตรวจสอบ HTTP Header ชื่อ Strict-Transport-Security ที่เป็นค่าระยะเวลาที่กำหนดการทำงานเพื่อบังคับใช้ HTTPS โดยเว็บเบราว์เซอร์จะบังคับใช้ HTTPS สำหรับเว็บไซต์นั้นๆ ตามระยะเวลาที่กำหนดใน HTTP Header โดยการใช้งาน HSTS รองรับการทำงานกับเบราว์เซอร์ Chrome และ Firefox เวอร์ชัน 4 ขึ้นไป และจำกัดการใช้งานสำหรับรายชื่อเว็บไซต์ที่ถูกติดตั้งใน HSTS List เท่านั้นเช่น paypal.com เป็นต้น สำหรับชื่อเว็บไซต์ที่ไม่ถูกติดตั้งใน HSTS List เมื่อต้องการใช้งาน เว็บเซิร์ฟเวอร์ต้องกำหนดการตอบกลับ HTTP Header และให้ผู้ใช้กำหนดชื่อเว็บไซต์บนเว็บเบราว์เซอร์เพื่อบังคับใช้ HTTPS

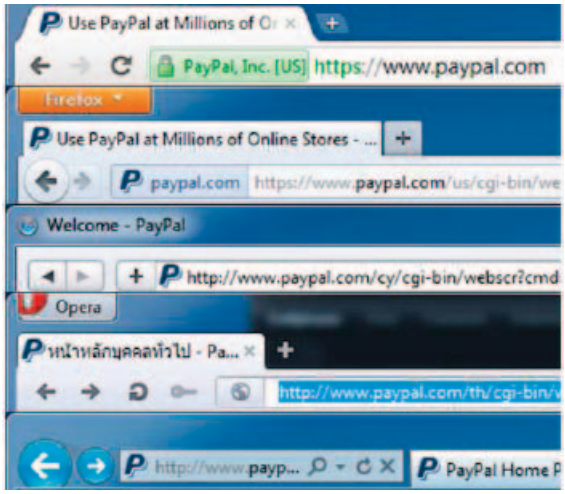
3. การทดสอบระบบป้องกันการโจมตี HTTPS

3.1 เกณฑ์การประเมิน

การทดสอบเพื่อประเมินผลประสิทธิภาพความมั่นคงในการป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS ใช้เครื่องมือ SSL Strip ในการทดสอบโจมตี เว็บเบราว์เซอร์ 5 ชนิด คือ Google Chrome 17, Fire-fox 10, Internet Explorer 9, Safari 5.1.2 และ Opera 11 โดยกำหนดเกณฑ์ในการประเมินคือ ระบบสามารถแจ้งเตือนหรือป้องกันการโจมตี ระบบง่ายต่อการใช้งานของผู้ใช้ ระบบง่ายต่อการปรับใช้กับระบบปัจจุบันเพื่อพัฒนาเว็บไซต์ให้ปลอดภัย และความสามารถในการรองรับการทำงานกับเว็บเบราว์เซอร์

3.2 ผลการทดสอบระบบที่เสนอก่อนหน้านี้

จากการศึกษางานวิจัยก่อนหน้านี้สำหรับระบบ HProxy, SSLock, HTTPSLock และ EV-SSL สามารถตรวจสอบแต่ไม่สามารถป้องกันการโจมตีด้วยวิธี SSL Strip และระบบ HSTS มีประสิทธิภาพในการป้องกันการโจมตี ดังนั้นในงานวิจัยนี้จึงได้ทดสอบโจมตี HSTS



รูปที่ 3 ผลการโจมตี HSTS ด้วยวิธี SSL Strip

ด้วย SSL Strip เพื่อวิเคราะห์ประสิทธิภาพการป้องกันโดยให้เหยื่อส่งการร้องขอโดยระบุชื่อเว็บไซต์ที่ถูกติดตั้งใน HSTS List คือ paypal.com ด้วยเว็บเบราว์เซอร์ 5 ชนิดคือ Google Chrome, Firefox, Internet Explorer, Safari และ Opera ผลการทดสอบโจมตีแสดงดังรูปที่ 3

จากรูปที่ 3 เป็นผลการทดสอบการทำงานของระบบ HSTS สามารถป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS ได้เฉพาะในเว็บเบราว์เซอร์ Google Chrome และ Firefox เท่านั้น และเมื่อเหยื่อเรียกใช้งานเว็บไซต์ผ่านเว็บเบราว์เซอร์อื่นแล้วส่งข้อมูลสำคัญ ชื่อบัญชีผู้ใช้ และรหัสผ่านไปที่เว็บเซิร์ฟเวอร์ผู้โจมตีสามารถดักจับข้อมูลสำคัญของเหยื่อได้ เนื่องจากการสื่อสารที่เครื่องเหยื่อดำเนินบนโพรโทคอล HTTP โดยสามารถสรุปผลการประเมินการทดสอบการใช้งาน และการโจมตีระบบที่เสนอในงานวิจัยก่อนหน้านี้เพื่อป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS ด้วยวิธีการโจมตีแบบ SSL Strip ดังตารางที่ 1

จากตารางที่ 1 เป็นผลการประเมินระบบป้องกันการโจมตี HTTPS แสดงให้เห็นว่า HProxy, SSLock, HTTPSLock และ EV-SSL สามารถแจ้งเตือนการโจมตีได้เท่านั้น อีกทั้ง HProxy ยังมีปัญหาความสะดวกในการใช้งานเนื่องจากต้องติดตั้งเครื่องมือบนเครื่องผู้ใช้ระบบ

ของ SSLock และ HTTPSLock มีปัญหาในการใช้งานที่ระบบไม่รองรับการทำงานกับทุกเว็บเบราว์เซอร์และไม่สามารถป้องกันการโจมตี

ตารางที่ 1 ผลการประเมินระบบป้องกันการโจมตี HTTPS

Solution	Protect	Usage Simplicity	Independency of Browser
HProxy	×	×	✓
SSLock	×	✓	×
HTTPSLock	×	✓	×
EV-SSL	×	✓	✓
HSTS	✓*	×	×

✓ *Support* × *Not support* * *Failed in some case*

จากผลการประเมิน HSTS เป็นวิธีเดียวที่สามารถป้องกันการโจมตีด้วยวิธี SSL Strip อย่างไรก็ตามการใช้งาน HSTS มีปัญหาในเรื่องความสะดวกต่อผู้ใช้ในการกำหนดชื่อเว็บไซต์ หากผู้ใช้ไม่ปฏิบัติตามขั้นตอนของ HSTS ระบบเว็บไซต์ก็สามารถถูกโจมตีด้วยวิธี SSL Strip อีกทั้งการทำงานของ HSTS รองรับบนเว็บเบราว์เซอร์ 2 ชนิดคือ Chrome และ Firefox ทำให้เกิดปัญหาในการใช้งานเนื่องจากในปัจจุบันมีเว็บเบราว์เซอร์จำนวนมากที่ต้องการความปลอดภัย

4. ระบบป้องกันการโจมตี HTTPS รูปแบบใหม่

จากการศึกษาพบว่า การทำงานของระบบเว็บไซต์ที่มีความมั่นคงในการใช้งาน ระบบต้องทำงานบนโพรโทคอล HTTPS เนื่องจากสามารถป้องกันการโจมตีเว็บไซต์ได้หลายรูปแบบโดยเฉพาะการโจมตีแบบ MITM ที่เป็นพื้นฐานของการโจมตีเว็บไซต์หลายรูปแบบ อย่างไรก็ตาม HTTPS สามารถโจมตีได้ด้วยวิธีการต่างๆ เช่น SSL Sniff และ SSL Strip

ดังนั้นในงานวิจัยนี้จึงเสนอระบบการตรวจสอบ และป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จากการโจมตีด้วยวิธี SSL Strip ที่เป็นปัญหาสำคัญต่อความมั่นคงในการใช้งานเว็บไซต์ เนื่องจากผลการโจมตีเว็บเบราว์เซอร์ไม่สามารถตรวจสอบและป้องกันการโจมตีซึ่งการสื่อสารข้อมูลบนเครื่องเหยื่อดำเนินบนโพรโทคอล HTTP ทำให้ผู้โจมตีสามารถดักจับข้อมูลสำคัญของเหยื่อและนำไปสู่การโจมตีระบบเว็บไซต์ด้วยวิธีต่างๆ เช่นการจู่โจมแบบเล่นซ้ำคุ้กกี้ และวิธีการจู่โจมแบบ Sidejacking โดยระบบที่เสนอคำนึงถึงประสิทธิภาพในการป้องกันเป็นวิธีการที่รองรับการทำงานกับทุกเว็บเบราว์เซอร์ ง่ายต่อการใช้งานของผู้ใช้ และการปรับใช้กับระบบเว็บไซต์โดยออกแบบและพัฒนาระบบต้นแบบในรูปของ API เพื่อง่ายต่อการพัฒนาเว็บไซต์ให้มีความมั่นคง โดยมีรายละเอียดดังต่อไปนี้

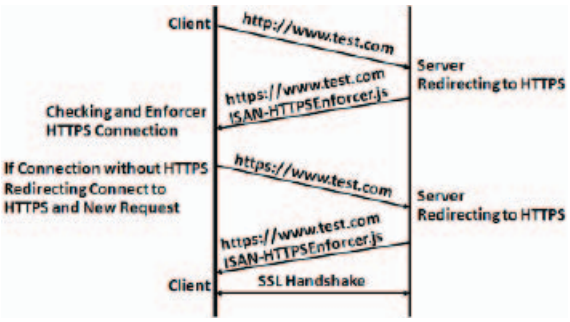
4.1 การออกแบบระบบตรวจสอบและบังคับใช้ HTTPS

จากรูปแบบการโจมตีเว็บไซต์ที่ทำงานบน HTTPS ด้วยการโจมตีแบบ SSL Strip ที่มีการเปลี่ยนแปลงโพรโทคอลในการสื่อสารบนเครื่องเหยื่อจาก HTTPS เป็น HTTP และการทำงานของเว็บเบราว์เซอร์ที่ไม่แสดงข้อความแจ้งเตือนเนื่องจากการสื่อสารดำเนินบน HTTP ปกติ ดังนั้นในงานวิจัยนี้จึงออกแบบระบบตรวจสอบ และป้องกันการโจมตีเว็บไซต์ โดยมีแนวคิดให้เว็บเบราว์เซอร์สามารถตรวจสอบ และบังคับใช้รูปแบบการสื่อสารกับเว็บเซิร์ฟเวอร์บนโพรโทคอล HTTPS โดยใช้ชื่อว่า ISAN-HTTPS Enforcer มีรายละเอียดดังต่อไปนี้

1. เมื่อผู้ใช้เรียกใช้งานเว็บไซต์ โดยปกติพฤติกรรมของผู้ใช้ในการระบุชื่อเว็บไซต์บนเว็บเบราว์เซอร์จะระบุชื่อเว็บไซต์เช่น test.com หรือ www.test.com โดยไม่มีการระบุโพรโทคอล เมื่อมีการส่งคำขอไปที่เว็บเซิร์ฟเวอร์ เว็บเบราว์เซอร์จะประมวลผลโดยเริ่มต้นการสื่อสารกับเว็บเซิร์ฟเวอร์บนโพรโทคอล HTTP

2. เมื่อการร้องขอมาถึงเว็บเบราว์เซอร์ก็จะประมวลผลโดยกำหนดการสื่อสารข้อมูลและตอบกลับมาที่เครื่องไคลเอนต์ด้วยโพรโทคอล HTTPS ซึ่งข้อมูลที่ตอบกลับมาที่เครื่องไคลเอนต์ประกอบด้วยเนื้อหาของเว็บไซต์และ ISAN-HTTPS Enforcer ที่พัฒนาโดยภาษา Javascript

3. เมื่อข้อมูลตอบกลับมาที่เครื่องไคลเอนต์การทำงานของ ISAN-HTTPS Enforcer จะประมวลผลเพื่อตรวจสอบและบังคับใช้โพรโทคอล HTTPS ในการสื่อสารระหว่างเว็บเบราว์เซอร์ กับเว็บเซิร์ฟเวอร์ดังรูปที่ 4



รูปที่ 4 การทำงานของ ISAN-HTTPS Enforcer

รูปที่ 5 การแก้ไขไฟล์ index.php เพื่อเรียกใช้ไฟล์ ISAN-HTTPSEnforcer.js

4.2 การพัฒนาระบบตรวจสอบและบังคับใช้ HTTPS

ISAN-HTTPS Enforcer ถูกพัฒนาให้อยู่ในรูปของ API ด้วยภาษา Javascript เพื่อความสะดวกในการพัฒนาเว็บไซต์ให้มีความมั่นคง ระบบที่ได้สามารถรองรับการทำงานกับทุกเว็บเบราว์เซอร์ และสามารถป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จากการโจมตีด้วยวิธี SSL Strip โดยที่ผู้ใช้ไม่จำเป็นต้องติดตั้งเครื่องมือหรือกำหนดการใช้งานบนเว็บเบราว์เซอร์เพื่อป้องกันการโจมตี เนื่องจากระบบถูกออกแบบให้มีการควบคุมการทำงานเพื่อบังคับใช้โพรโทคอล HTTPS ในการสื่อสารบนเว็บเซิร์ฟเวอร์ โดยตัวอย่างการเรียกใช้งาน API เพื่อติดตั้งบนโปรแกรม Moodle จะต้องแก้ไขไฟล์ index.php เพื่อเรียกใช้ไฟล์ ISAN-HTTPSEnforcer.js ที่มีหน้าที่ในการตรวจสอบและบังคับใช้โพรโทคอล HTTPS ดังรูปที่ 5 หมายเลข 1 และหมายเลข 2 เป็นการเปิดใช้งานโพรโทคอล HTTPS เพื่อใช้ในการสื่อสารด้วยภาษา PHP

5. การทดสอบระบบ

การทดสอบระบบเพื่อเปรียบเทียบประสิทธิภาพในการป้องกันการโจมตี และประสิทธิภาพในการใช้งานวิธีการป้องกันเว็บไซต์ที่ทำงานบน HTTPS โดยทดสอบที่เครื่องเซิร์ฟเวอร์ซึ่งมีรายละเอียดของเซิร์ฟเวอร์ คือใช้ Intel® XEON™ CPU 2.40 GHz RAM 2 GB ระบบปฏิบัติการ CentOS 5.3 เว็บเซิร์ฟเวอร์ Apache/2.2.0 PHP Version 5.2.4 ฐานข้อมูล MySQL Version 5.0.24a และเครื่องไคลเอนต์ใช้ CPU INTEL CORE 2 QUAD Q6600 RAM 4 GB ระบบปฏิบัติการ Windows XP เบราว์เซอร์ Google Chrome 17, Fire-fox 10, Internet Explorer 9, Safari 5.1.2 และ Opera 11 ในหน่วยปฏิบัติการวิจัย Information Security & Advanced Network (ISAN) โดยทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการป้องกันการโจมตีเว็บไซต์ด้วยวิธี SSL Strip และเพื่อเปรียบเทียบความยากง่ายในการใช้งาน โดยการวัดค่าเวลารวมในการทำกิจกรรมต่างๆ บนเว็บเบราว์เซอร์ด้วยวิธี Keystroke Level Model (KLM) [15] เพื่อเปรียบเทียบระหว่างรูปแบบการป้องกันที่เสนอในงานวิจัยนี้กับวิธีการของ HSTS โดยมีรายละเอียดในการทดสอบดังต่อไปนี้

5.1 การทดสอบประสิทธิภาพการป้องกันการโจมตี

การทดสอบประสิทธิภาพของ ISAN-HTTPS Enforcer ในการป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS ด้วยเว็บเบราว์เซอร์ 20 ชนิด โดยทดสอบใช้งานเว็บไซต์ที่ติดตั้งระบบ ISAN-HTTPS Enforcer และเว็บไซต์ที่ติดตั้งใน HSTS List จำนวน 100 ครั้ง ในขณะที่ผู้โจมตีใช้วิธีการโจมตีแบบ SSL Strip พบว่า ISAN-HTTPS Enforcer สามารถป้องกันการโจมตีเว็บไซต์โดยที่เว็บเซิร์ฟเวอร์ตอบกลับข้อมูลบนโพรโทคอล HTTPS ที่แถบระบุชื่อเว็บไซต์บนเว็บเบราว์เซอร์แสดงข้อมูลการใช้งานเว็บไซต์บนโพรโทคอล HTTPS และ EV-SSL ทั้งหมดจำนวน 100 ครั้ง ดังแสดงผลการทดสอบในตารางที่ 2

ตารางที่ 2 การทดสอบประสิทธิภาพของระบบป้องกันการโจมตี HTTPS

Web Browser	HSTS	ISAN-HTTPS Enforcer
Internet Explorer	×	✓
Firefox	✓*	✓
Google Chrome	✓	✓
Safari	×	✓
Opera	×	✓
SeaMonkey	×	✓
K-Meleon	×	✓
Konqueror	×	✓
Netscape Navigator	×	✓
Maxthon	×	✓
Lunascape 6	×	✓
SlimBrowser	×	✓
Arora	×	✓
Enigma Browser	×	✓
Opera Mini	×	✓
Skyfire	×	✓
Crazy	×	✓
Chromium	×	✓
Wyzo Media Browser	×	✓
WebKit-Based	×	✓

✓ *Protect* × *Not protect* * *Failed in some case*

จากตารางที่ 2 จะเห็นได้ว่าการทำงานของระบบ HSTS สามารถป้องกันการโจมตีด้วยวิธี SSL Strip บนเว็บเบราว์เซอร์ 2 ชนิดคือ Google Chrome และใน Firefox สามารถป้องกันการโจมตีได้ในกรณีผู้ใช้เรียกใช้งานเว็บไซต์บน Firefox เวอร์ชัน 4 ขึ้นไป ส่วนการทำงานของ ISAN-HTTPS Enforcer รองรับการทำงานกับทุกเว็บเบราว์เซอร์และมีประสิทธิภาพในการป้องกันการโจมตีเว็บไซต์จากการโจมตีด้วยวิธี SSL Strip

5.2 การทดสอบประสิทธิภาพการใช้งาน

ในการทดสอบประสิทธิภาพการใช้งานวิธีการที่เสนอเพื่อแก้ไขปัญหาการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จากวิธีการโจมตีแบบ SSL Strip โดยเปรียบเทียบการใช้งานระหว่างระบบ ISAN-HTTPS Enforcer กับวิธีการกำหนดการบังคับใช้ HTTPS ของ HSTS บนเว็บเบราว์เซอร์ Google Chrome 17 ดังรูปที่ 6 โดยการทดสอบเพื่อประเมินเวลาในการทำกิจกรรมต่างๆ บนเว็บเบราว์เซอร์ใช้วิธี KLM ในการทดสอบ กำหนดชื่อเว็บไซต์คือ test.com โดยผลการทดสอบแสดงเวลาทั้งหมดที่ใช้ในการทำกิจกรรมต่างๆ บนเว็บเบราว์เซอร์ หน่วยเป็นวินาที (second) ดังตารางที่ 3

รูปที่ 6 ตัวอย่างการกำหนดชื่อเว็บไซต์ใน HSTS List

ตารางที่ 3 ผลการทดสอบโดยใช้วิธีการ KLM

Solution	Total (seconds)
ISAN-HTTPS Enforcer	3.30
HSTS	20.45

จากการทดสอบประสิทธิภาพการใช้งานโดยการวัดค่าด้วยวิธี KLM พบว่าเวลาในการใช้งาน ISAN-HTTPS Enforcer น้อยกว่าการใช้งานวิธีการป้องกันของระบบ HSTS แสดงให้เห็นว่า ISAN-HTTPS Enforcer ที่พัฒนาให้อยู่ในรูปของ API เพื่อความสะดวกในขั้นตอนการพัฒนาเว็บไซต์ สามารถเพิ่มความมั่นคงให้กับระบบเว็บไซต์ที่ทำงานบน HTTPS โดยที่ผู้ใช้ไม่ต้องกำหนดการทำงานบนเว็บเบราว์เซอร์เนื่องจากมีการกำหนดการบังคับใช้ HTTPS ในการสื่อสารโดยเว็บเซิร์ฟเวอร์ อีกทั้งรองรับการทำงานกับทุกเว็บเบราว์เซอร์ และมีประสิทธิภาพในการป้องกันการโจมตีเว็บไซต์ที่ทำงานบน HTTPS จากการโจมตีด้วยวิธี SSL Strip โดยสรุปผลการทดสอบดังตารางที่ 4

ตารางที่ 4 สรุปผลการทดสอบระบบป้องกันการโจมตี HTTPS

Solution	Protect	Usage Simplicity	Independency Browser
HProxy	×	×	✓
SSLock	×	✓	×
HTTPSLock	×	✓	×
EV-SSL	×	✓	✓
HSTS	✓*	×	×
ISAN-HTTPS En-forcer	✓	✓	✓

✓ *Protect* × *Not protect* * *Failed in some case*

6. สรุป

จากการออกแบบการประเมินระบบของงานวิจัยก่อนหน้านี้ที่เสนอวิธีการเพื่อแก้ไขปัญหาการโจมตีเว็บไซต์ที่ทำงานบนโพรโทคอล HTTPS พบว่าการทำงานของระบบในงานวิจัยก่อนหน้านี้ไม่สามารถป้องกันการโจมตีเว็บไซต์จากการโจมตีด้วยวิธี MITM และการโจมตีด้วยวิธี SSL Strip ปัญหาความสะดวกในการใช้งาน ปัญหาความสามารถในการปรับใช้กับระบบปัจจุบันเนื่องจากระบบไม่ถูกพัฒนาให้เป็นมาตรฐาน รวมถึงปัญหาระบบไม่รองรับการทำงานกับทุกเว็บเบราว์เซอร์ ในงานวิจัยนี้จึงได้เสนอระบบป้องกันการโจมตีเว็บไซต์จากการโจมตีแบบ MITM และการโจมตีแบบ SSL Strip โดยเสนอ ISAN-HTTPS Enforcer ที่พัฒนาให้อยู่ในรูปของ API เพื่อความสะดวกในการพัฒนาเว็บไซต์ให้มีความมั่นคง สามารถตรวจสอบและบังคับใช้ HTTPS ในการสื่อสารโดยเว็บเซิร์ฟเวอร์เพื่อความสะดวกในการใช้งานเว็บไซต์ ซึ่งจากผลการทดสอบแสดงให้เห็นว่าระบบที่เสนอมีความสะดวกในการใช้งานโดยวัดจากเวลาทั้งหมดในการทำกิจกรรมต่างๆ บนเว็บเบราว์เซอร์ด้วยวิธี KLM ระบบที่เสนอรองรับการทำงานกับทุกเว็บเบราว์เซอร์และง่ายต่อการปรับใช้กับระบบเว็บไซต์เนื่องจากพัฒนาให้อยู่ในรูปของ API ด้วยภาษา Javascript รวมถึงระบบมีประสิทธิภาพในการป้องกันการโจมตีเว็บไซต์จากการโจมตีแบบ MITM ที่นำไปสู่การจู่โจมแบบเล่นซ้ำคุ้กกี้ การจู่โจมแบบ Side-jacking และการโจมตีด้วยวิธี SSL Strip

7. กิตติกรรมประกาศ

คณะผู้วิจัยขอขอบคุณ มหาวิทยาลัยมหาสารคามและ วช ที่ให้การสนับสนุนทุนวิจัย คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคามให้การสนับสนุนสถานที่ในการดำเนินงาน ตลอดจนอุปกรณ์เครือข่ายในการทดลองและวิจัยในครั้งนี้

เอกสารอ้างอิง

[1] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, and P. Leach, "Hypertext Transfer Protocol -- HTTP/1.1," *IETF*, *RFC 2616*, June 1999.

[2] M. Howard, "Man-in-the-Middle Attack to the HTTPS Protocol," *IEEE computer society*, pp. 78-81, February 2009.

[3] R. Graham, "SideJacking with Hamster." [Online]. Available: http://erratasec.blogspot.com/2007/08/sidejacking-with-hamster_05.html.

[4] E. Rescorla, "HTTP Over TLS," *IETF*, RFC 2818, May 2000.

[5] T. Dierks and C. Allen, "The TLS Protocol," *IETF*, RFC 2246, December 1999.

[6] M. Marlingspike, "New Tricks For Defeating SSL in Practice," *BlackHat Conference*, USA, 2009.

[7] N. Sriwiboon and S. Puangpronpitag, "Security Performance Evaluation of Solutions on Cookie Attacks," *in National Conference on Information Technology (NCIT)*, Thailand, April 2012.

[8] C. Soghoian and S. Stamm, "Certified Lies: Detecting and Defeating Government Interception Attacks against SSL," In G. *Danezis (Ed.)* Proceedings of the Fifteenth International Conference on Financial Cryptography and Data

Security, St. Lucia, February 2011.

[9] K. Cheng, M. Gao, and R. Guo, "Analysis and Research on HTTPS Hijacking Attacks," *Proceedings of the 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing* – Vol. 02, IEEE Computer Society Washington, DC, USA, April 2010.

[10] N. Nikiforakis, Y. Younan, and W. Joosen, "HProxy: Client-side detection of SSL stripping attacks," in *Proceedings of the 7th Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2010.

[11] A. Fung and K. Cheung, "SSLock: Sustaining the Trust on Entities Brought by SSL," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Beijing, China, April 2010.

[12] A. Fung and K. Cheung, "HTTPSLock: Enforcing HTTPS in Unmodified Browsers with Cached Javascript," in *Proceedings of the Network and System Security (NSS), 4th International Conference*, Melbourne, Victoria Australia, September 2010.

[13] J. Hodges, C. Jackson, and A. Barth, "HTTP Strict Transport Security (HSTS)." [Online]. Available : http://tools.ietf.org/html/draft-ietf-websec-strict-transport -sec-04, January 2012.

[14] C. Jackson and A. Barth, "ForceHTTPS: Protecting High-Security Web Sites from Network Attacks," in *Proceedings of the International World Wide Web Conference (WWW)*, 2008.

[15] S. Card, T. Moran T, A. Newell, "The Keystroke-Level Model for User Performance Time with Interactive Aystems," *ACM Communation Journal*, vol. 37, no.7, pp.396-410, 1980.