การตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่

นายวรวิชญวิทย์ ประเสริฐยิ่ง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

DETECTION OF CROSS-SITE SCRIPTING USING WEB PROXY

Mister Worawitchayawit Prasoetying

A Thesis Submitted in Partiral Fulfuillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2012

หัวข้อวิทยานิพนธ์	การตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่
โดย	นายวรวิชญวิทย์ ประเสริฐยิ่ง
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.ยรรยง เต็งอำนวย)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.พงศ์ธวัช ชีพพิมลชัย)

วรวิชญวิทย์ ประเสริฐยิ่ง : การตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่. (DETECTION OF CROSS-SITE SCRIPTING USING WEB PROXY) อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ. ดร.เกริก ภิรมย์โสภา, 43 หน้า.

งานวิจัยชิ้นนี้นำเสนอเกี่ยวกับวิธีการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซีในขณะที่เว็บไซต์ใช้งานจริง ณ ฝั่งเครื่องคอมพิวเตอร์ผู้ใช้งาน ซึ่งปัญหาครอสไซต์สคริปติ้งเป็น 1 ใน 3 ปัญหาหลักของเว็บระบบประยุกต์ โดยวิธีการตรวจจับและแก้ปัญหาส่วนใหญ่มักจะกระทำที่ฝั่งเครื่องบริการ แต่ปกติผู้ใช้งานทั่วไปจะไม่มีสิทธิ์ในการแก้ไขใดๆ บนฝ่ายเครื่องบริการ วัตถุประสงค์หลักในการทำงานระบบประยุกต์นี้คือแสดงผลความเสี่ยงของปัญหาครอสไซต์สคริปติ้ง เพื่อเป็นข้อมูลเพื่อให้ผู้ใช้งานตัดสินใจได้ว่าจะเข้าใช้งานเว็บระบบประยุกต์ดังกล่าวหรือไม่ ซึ่งผลลัพธ์ที่ได้จากระบบประยุกต์นั้นจะต้องเสียเวลาเพิ่มเติมในการตรวจสอบ

ภาควิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา ..วิทยาศาสตร์คอมพิวเตอร์.. ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา .....2555.....

5470357321 : MAJOR COMPUTER SCIENCE
KEYWORDS: CROSS-SITE SCRIPTING / WEB PROXY / DETECTION
WORAWITCHAYAWIT PRASOETYING : DETECTION OF CROSS-SITE SCRIPTING USING WEB PROXY. ADVISOR : ASST. PROF. KRERK PIROMSOPA, Ph.D., 43 pp.

This research proposes the detection of cross-site scripting in web application by using web proxy during run-time on a client side. Cross-site scripting is the top three security problems of web application. Most solutions detected and solved the problem on a server side. However, users do not have authority to edit anything on the server. Proposed application shows the risk of cross-site scripting. Users have to decide to agree or to ignore, in order to visit the web application. The experiment verifies that the application detects interesting patterns with few overhead.

Department : ..COMPUTER ENGINEERING.. Student's Signature : .....
Field of Study : .....COMPUTER SCIENCE..... Advisor's Signature : .....
Academic year : .....2012.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ด้วยความอนุเคราะห์อย่างยิ่งของผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา อาจารย์ที่ปรึกษา ซึ่งท่านได้เสียสละและอุทิศเวลาเป็นจำนวนมาก ในการมอบความรู้ความเข้าใจ แนะนำแนวทางการทำวิจัย ให้แนวคิดในการทำงานและการใช้ชีวิต ตรวจสอบติดตามผล ปรับปรุงแก้ไข ให้คำแนะนำในทุกเรื่องอย่างวิริยะอุสาหะ และให้การสนับสนุนในด้านต่างๆ เป็นอย่างดียิ่ง จนทำให้การวิจัยในครั้งนี้ประสบผลสำเร็จออกมาด้วยดี

ขอขอบพระคุณ อาจารย์ ดร.ยรรยง เต็งอำนวย ประธานคณะกรรมการสอบวิทยานิพนธ์ และการเป็นบุคคลต้นแบบในการใช้ชีวิตด้านต่างๆ ทั้งด้านวิชาการ ด้านการทำวิจัย ด้านความสนใจเฉพาะทาง การใช้ชีวิตประจำวัน รวมไปถึงทุกเรื่องที่อาจารย์เมตตาให้คำสอน แนะนำด้วยความรักและห่วงใยอย่างแท้จริงเสมอมา

ขอขอบพระคุณ ดร.พงศ์ธวัช ชีพพิมลชัย และผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์ กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงลงได้

ขอขอบคุณ นายสมิทธ์ ธรรมบำรุง, นายปริน เจียมอนันตพงศ์, นายจักรรินทร์ เทิดภาปิยะนาค, นายสุวัจชัย ตั้งเผ่าพงศ์, นายสิทธิโชค แสงไกรรุ่งเรือง และนายพสิษฐ์ คงคุณากรกุล ที่ได้คอยให้ความช่วยเหลือต่างๆ ในด้านวิชาการ การพัฒนาระบบ การทำวิจัย แนะนำและช่วยแก้ปัญหาที่พบ ขอขอบคุณ นางสาวประภาวดี เอกวงศ์ และนายกฤษฎ์ สุวรรณภูมิ ตลอดจนขอบคุณเพื่อนๆ พี่ๆ น้องๆ รวมไปถึงเจ้าหน้าที่ ในห้องปฏิบัติการวิจัยระบบ แพลตฟอร์ม และสถาปัตยกรรม (SPALAB), สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์รุ่นที่ 41 (CS41), ภาควิชาวิศวกรรมคอมพิวเตอร์ (CPCU), ชมรมบัณฑิตศึกษา คณะวิศวกรรมศาสตร์ (EGSACU), คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่คอยให้ความช่วยเหลือและให้กำลังใจอย่างดีเรื่อยมา

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ บุญชัย โสวรรณวณิชกุล ที่ได้เมตตาอนุเคราะห์ทุนการศึกษาในระดับบัณฑิตศึกษา และคอยให้คำปรึกษา รวมไปถึงให้การสนับสนุนในด้านต่างๆ ด้วยความกรุณาเสมอมา

ท้ายที่สุด ผู้เสนอวิทยานิพนธ์ขอขอบคุณครอบครัว รวมไปถึงพี่ๆ น้องๆ และเพื่อนทุกๆ คน ที่คอยติดตาม ให้กำลังใจและสนับสนุน รวมทั้งท่านอื่นๆ ที่มิได้กล่าวชื่อไว้ ณ ที่นี้ที่มีส่วนช่วยให้วิทยานิพนธ์สำเร็จได้ด้วยดี

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญรูป

หน้า

บทที่ 1
บทนำ

ในบทนำนี้จะแบ่งเป็นเจ็ดหัวข้อย่อย กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขตของการวิจัย ประโยชน์ที่คาดว่าจะได้รับ วิธีดำเนินการวิจัย ลำดับขั้นตอนในการเสนอผลการวิจัย และผลงานที่ตีพิมพ์จากวิทยานิพนธ์ ตามลำดับ ดังนี้

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันปัญหาการโจมตีเว็บระบบประยุกต์ (Web application) มีจำนวนมากขึ้นเรื่อยๆ โดยปัญหาที่พบว่าเกิดขึ้นมากอย่างหนึ่งก็คือปัญหาของครอสไซต์สคริปติ้ง (Cross-site scripting) เนื่องจากการโจมตีชนิดนี้สามารถทำได้ง่าย จากข้อมูลของ“The Open Web Application Security Project (OWASP)” เมื่อปี ค.ศ. 2013 ได้ระบุเอาไว้ว่าปัญหาเรื่องของครอสไซต์สคริปติ้งมีมากเป็นอันดับ 3 (ดังข้อมูลในตารางที่ 1) จากปัญหาที่พบบ่อยด้านความปลอดภัยและช่องโหว่ [1] และจากรายงานต่อสาธารณชนพบว่าในหลายปีที่ผ่านมาปัญหาของครอสไซต์สคริปติ้งถูกพบบ่อยมากขึ้นจนแซงปัญหาบัฟเฟอร์โอเวอร์โฟลว์เลยทีเดียว[2]

ตารางที่ 1 ตารางแสดงปัญหาการโจมตีเว็บระบบประยุกต์

OWASP Top 10 – 2013
1 – Injection
2 – Broken Authentication and Session Management
3 – Cross-Site Scripting (XSS)
4 – Insecure Direct Object References
5 – Security Misconfiguration
6 – Sensitive Data Exposure
7 – Missing Function Level Access Control
8 – Cross-Site Request Forgery (CSRF)
9 – Using Known Vulnerable Components
10 – Unvalidated Redirects and Forwards

เทคนิคครอสไซต์สคริปติ้งเป็นเทคนิคโจมตีที่เครื่องคอมพิวเตอร์ผู้ใช้งาน (Client computer) สามารถส่งข้อมูลที่รับเข้า (Input) ไปยังเครื่องบริการ (Server) ซึ่งเครื่องบริการไม่ได้ทำการตรวจสอบความปลอดภัยของข้อมูลที่รับเข้ามาแล้วนำไปทำการประมวลผลเป็นข้อมูลที่ส่งออก (Output) ส่งไปแสดงผลไปยังเบราว์เซอร์ที่เครื่องคอมพิวเตอร์ผู้ใช้งาน โดยที่ผู้โจมตี (Attacker) จึงอาศัยช่องโหว่นี้ในการทำการโจมตี โดยเขียนบทคำสั่ง (Script) แล้วทำการส่งไปยังเครื่องบริการ

ปัจจุบันผู้จัดทำเว็บระบบประยุกต์ส่วนใหญ่นิยมทำการทดสอบเบื้องต้นในปัญหาครอสไซต์สคริปติ้ง [3][4] ก่อนให้บุคคลทั่วไปเข้าใช้งาน แต่ก็ยังมีเว็บไซต์จำนวนมากที่ไม่ได้ทำการตรวจสอบความปลอดภัยในเรื่องดังกล่าว ทำให้ผู้เข้าใช้งานเจอปัญหาครอสไซต์สคริปติ้งและไม่สามารถตรวจจับหรือทราบได้ว่าตนเองกำลังถูกโจมตีด้วยเทคนิคครอสไซต์สคริปติ้ง โครงงานวิจัยนี้จึงถูกจัดทำขึ้นเพื่อช่วยตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่ (Web proxy) ซึ่งคาดว่าจะสามารถช่วยทำการตรวจจับเว็บไซต์ที่ใช้งานได้เบื้องต้นว่ามีปัญหาครอสไซต์สคริปติ้ง และผู้เข้าใช้งานสามารถตัดสินใจได้ว่าสมควรเข้าใช้งานเว็บระบบประยุกต์ดังกล่าวต่อไปหรือไม่ได้ด้วยตนเอง

หากวิธีการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่ (Detection of Cross-Site Scripting using web proxy) นี้ สามารถทำการตรวจจับเว็บระบบประยุกต์ที่มีปัญหาครอสไซต์สคริปติ้งได้จริงดังที่คาดหวังไว้ จะช่วยแจ้งเตือนให้ผู้เข้าใช้งานรับทราบและปิดช่องโหว่ของปัญหาได้จำนวนมาก และทำให้ผู้เข้าใช้งานเว็บระบบประยุกต์มีความปลอดภัยมากขึ้นทีเดียว

1.2 วัตถุประสงค์ของการวิจัย

การวิจัยมีวัตถุประสงค์ ดังนี้

1) เพื่อศึกษาหาวิธีการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่

2) เพื่อนำวิธีการตรวจจับนี้มาประยุกต์ใช้จริง

3) เพื่อทดลองหารูปแบบของปัญหาครอสไซต์สคริปติ้งที่วิธีการตรวจจับดังกล่าวสามารถตรวจจับได้ และไม่สามารถตรวจจับได้

4) เพื่อวัดและวิเคราะห์ประสิทธิภาพของเทคนิคการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่

1.3 ขอบเขตของการวิจัย

ขอบเขตของการวิจัยถูกกำหนดไว้ ดังนี้

1) เลือกเฉพาะวิธีการป้องกันปัญหาครอสไซต์สคริปติ้งจากวิธีป้องกันฝั่งผู้ใช้งาน (Client-side prevention) เท่านั้นในการทดลอง ดังตารางที่ 2

ตารางที่ 2 ตารางแสดงการเปรียบเทียบวิธีการแก้ไขปัญหาครอสไซต์สคริปติ้ง

Method	Code modification	User involvement	Applicable before deployment	Generate concrete attack	Locate vulnerability	Input source ID	Runtime overhead	XSS exploits addressed
Defensive coding	Yes	Intensive	Yes	Not applicable	Not applicable	Not applicable	Not applicable	All types
Input validation testing	No	Intensive	Yes	Yes	Not explicitly	Yes	No	All types
Fault-based XSS testing	Yes	Intensive	Yes	Yes	Yes	Yes	No	All types
Static analysis	No	Average	Yes	No	Yes	Yes	No	Reflected and stored
Static string analysis	No	Low	Yes	Not explicitly	Yes	Yes	No	Reflected and stored
Combined static and dynamic analysis	No	Low	Yes	Yes	Yes	Yes	No	Reflected and stored
Server-side prevention	Yes	Average	No	No	No	No	Yes	All types
Client-side prevention	No	Intensive	No	No	No	No	Yes	All types

2) งานวิจัยนี้จะศึกษาและทดลองการตรวจจับปัญหาครอสไซต์สคริปติ้งแบบถาวร (Stored Cross-Site Scripting) และครอสไซต์สคริปติ้งแบบชั่วคราว (Reflected Cross-Site Scripting) เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย ได้แก่

1) เข้าใจวิธีการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่

2) ได้ความรู้เกี่ยวกับรูปแบบการโจมตีที่วิธีการตรวจจับดังกล่าวสามารถตรวจจับได้และไม่สามารถตรวจจับได้ พร้อมเหตุผล

3) ได้รับความรู้เกี่ยวกับรูปแบบวิธีในการตรวจจับปัญหาครอสไซต์สคริปติ้ง

4) สามารถนำความรู้จากผลการวิจัยนี้ไปประยุกต์ใช้จริงต่อไปในอนาคต

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

วิธีดำเนินการวิจัย ถูกแบ่งเป็นห้าขั้นตอน ดังนี้

1) ศึกษาเทคโนโลยีและงานวิจัยที่เกี่ยวข้อง

2) ขั้นเตรียมเว็บพร็อกซี่เพื่อใช้ตรวจจับการโจมตี

3) ขั้นทดลองโจมตีด้วยการโจมตีแบบครอสไซต์สคริปติ้ง

4) ขั้นทดลองวัดประสิทธิภาพ

5) ขั้นสรุปผลการทดลองและจัดทำวิทยานิพนธ์

1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 5 บท ดังต่อไปนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึง ความเป็นมาและความสำคัญของปัญหา รวมถึงวัตถุประสงค์ของการวิจัย บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับการวิจัยนี้ บทที่ 3 การออกแบบระบบในงานวิจัย บทที่ 4 การสร้างและการทดลองระบบ และบทที่ 5 เป็นบทสรุปของงานวิจัย

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตอบรับให้ตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “Detection of Cross-Site Scripting using web proxy” โดยนายวรวิชญวิทย์ ประเสริฐยิ่ง และผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา, ในงานประชุมวิชาการ “The International Conference on E-Technologies and Business on the Web (EBW2013)” ณ มหาวิทยาลัยหอการค้าไทย กรุงเทพมหานคร ประเทศไทย วันที่ 7-9 พฤษภาคม พ.ศ. 2556

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

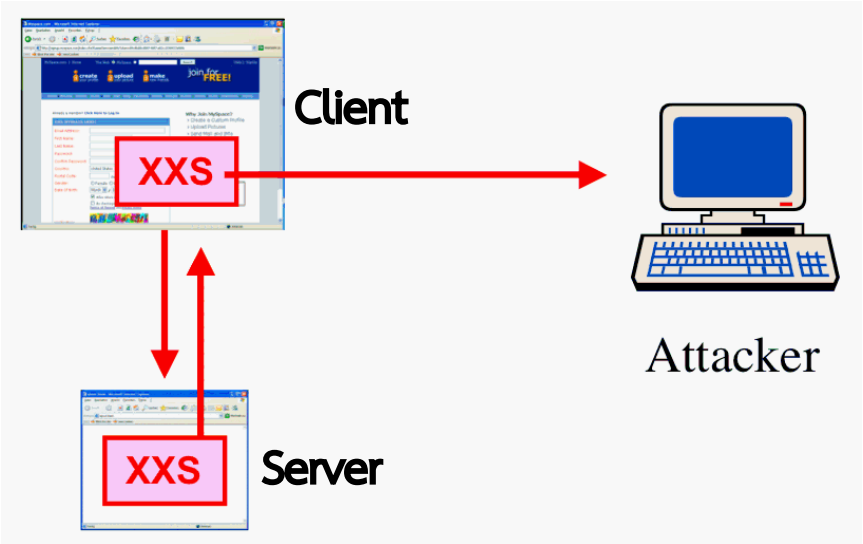
ในบทนี้จะกล่าวถึงแนวคิดและทฤษฎี รวมทั้งเอกสารและงานวิจัยที่เกี่ยวข้อง ดังนี้

2.1 แนวคิดและทฤษฎี

แนวคิดและทฤษฎีที่จะอธิบายในการวิจัยนี้ แบ่งเป็นสี่ส่วน ได้แก่

2.1.1 ครอสไซต์สคริปติ้ง (Cross-site scripting) [5]

ครอสไซต์สคริปติ้ง (Cross-site scripting : XSS) เป็นการโจมตีแบบระบบประยุกต์ (Application) ประเภท malicious injection โดยอาศัยหลักการที่คอมพิวเตอร์ผู้ใช้งาน (Client computer) สามารถส่งข้อความอะไรก็ได้ไปยังเครื่องบริการ (Server) แล้วเครื่องบริการนำข้อความที่ได้รับมาไปประมวลผล โดยที่เครื่องบริการไม่ได้ตรวจสอบข้อมูลที่รับเข้า (Input) และข้อมูลที่ส่งออก (Output) ว่ามีความปลอดภัยเพียงพอ ผู้โจมตี (Attacker) จึงอาศัยช่องโหว่นี้ในการโจมตี โดยการสร้างเป็นคำสั่ง (Code) แล้วส่งไปยังเครื่องบริการ แล้วเครื่องบริการก็จะทำการประมวลผลแล้วส่งผลตอบสนองมายังเบราว์เซอร์ (Browser) ของเครื่องคอมพิวเตอร์ผู้ใช้งานในรูปแบบบทคำสั่งฝ่ายผู้ใช้งาน (Client side script) ซึ่งบทคำสั่งฝ่ายผู้ใช้งานที่เกิดขึ้นนั้นมีสิทธิ์ในการทำงานเทียบเท่ากับบทคำสั่ง (Script) ที่เกิดขึ้นจากตัวเว็บไซต์เอง



รูปที่ 1 การโจมตีโดยวิธีการครอสไซต์สคริปติ้ง

ผู้โจมตีใช้เทคนิคครอสไซต์สคริปติ้งเพื่อข้อมูลหรือความสามารถบางอย่าง อาทิ เพื่อทำการส่งค่าคุกกี้ (Cookie) จากเครื่องคอมพิวเตอร์ผู้ใช้งานไปยังผู้โจมตี หรือเพื่อให้เครื่องคอมพิวเตอร์ผู้ใช้งานไปทำการบรรจุลง (Download) บทคำสั่งมาจากเว็บไซต์เพื่อให้สามารถใช้ทรัพยากรบางอย่างได้อย่างเต็มที่ เป็นต้น ทั้งนี้ขึ้นอยู่กับเทคโนโลยีของบทคำสั่งฝ่ายผู้ใช้งาน และเป้าหมายการโจมตีของผู้โจมตีด้วย

ครอสไซต์สคริปติ้งสามารถเกิดขึ้นได้ในหลายภาษา อาทิ HTML, Java Script, VB Script, ActiveX, Flash เป็นต้น แต่ภาษาที่นิยมใช้ครอสไซต์สคริปติ้งกันอย่างแพร่หลายอย่างยิ่ง คือ Java Script

ประเภทของครอสไซต์สคริปติ้ง

1) ครอสไซต์สคริปติ้งแบบถาวร (Stored Cross-Site Scripting)

ครอสไซต์สคริปติ้งแบบนี้เป็นเทคนิคการโจมตีแบบถาวร (Persistent) โดยจะอาศัยประโยชน์จากเว็บบอร์ด (Web board) เว็บบล็อก (Web blog) หรือสมุดเยี่ยมชม (Guestbook) ที่โดยปกติเว็บระบบประยุกต์ (Web application) กลุ่มนี้จะให้ผู้ใช้งานสามารถกรอกเนื้อหา (Content) ได้ด้วยตนเองหรือจะนำ ข้อมูลที่รับเข้าไปเก็บและใช้งานโดยที่ไม่ได้ทำการตรวจสอบความปลอดภัยก่อน ผู้โจมตีก็จะสามารถใส่บทคำสั่งเข้าไปได้โดยตรงเลย เช่น

```
<script>document.location='http://attacker.abc/cookie-steal.cgi?'+document.cookie</script>
```

ถ้าเป็นเว็บระบบประยุกต์ที่ไม่ได้มีการตรวจสอบเนื้อหาหรือข้อมูลที่รับเข้าไปเก็บไว้แล้ว เมื่อมีผู้เข้าชมคนอื่นเข้ามาเยี่ยมชมเว็บไซต์ที่มีบทคำสั่งของผู้โจมตี เครื่องคอมพิวเตอร์ของผู้เข้าชมคนนั้นก็จะทำการประมวลผลบทคำสั่งไปโดยอัตโนมัติ แล้วทำการส่งค่าคุกกี้ไปยังเครื่องบริการของผู้โจมตี

2) ครอสไซต์สคริปติ้งแบบชั่วคราว (Reflected Cross-Site Scripting)

ครอสไซต์สคริปติ้งแบบนี้เป็นเทคนิคการโจมตีแบบชั่วคราว (Non-persistent) โดยส่วนใหญ่จะอยู่ในรูปแบบของการเชื่อมโยง (Link) เมื่อเหยื่อ (Victim) ได้ทำการคลิกที่การเชื่อมโยงแล้วบทคำสั่งจะทำการโจมตี ปกติจะอยู่ในเว็บไซต์ที่มีการ รับข้อมูลที่รับเข้าจากผู้ใช้งานมาแสดงผลบนเบราว์เซอร์

ตัวอย่าง : ในเว็บระบบประยุกต์ที่มีกระบวนการทำงานตามคำสั่งที่ได้ร้องขอ เช่น เมื่อพิมพ์คำสั่ง

http://www.portal.abc/index.php/sessionid=12345678
&username=yok

แล้วเว็บไซต์จะแสดงผล “Welcome yok” ออกมาให้เห็นผ่านทางเบราว์เซอร์ โดยเว็บระบบประยุกต์นำข้อมูลที่รับเข้าจาก username มาทำการแสดงผลโดยตรง ซึ่งหากไม่มีการป้องกันก่อนที่จะนำมาแสดงผลก็จะเป็นจุดอ่อนให้ใช้เทคนิคครอสไซต์สคริปติ้งได้ ถ้าผู้โจมตีทำการสร้างการเชื่อมโยงแล้วส่งไปทางอีเมลให้เหยื่อทำการคลิกการเชื่อมโยงเหยื่อก็จะถูกทำการโจมตี

ตัวอย่าง : ในการเชื่อมโยงที่มีการใช้เทคนิคครอสไซต์สคริปติ้งแอบแฝงเพิ่ม บทคำสั่งเข้าไป แล้วทำการส่งอีเมลไปหาเหยื่อ

<a herf=“http://www.portal.abc/index.php/
sessionid=12345678&username=<script>
document.location=‘http://attacker.abc/
cookie-steal.cgi?’+document.cookie</script>”>
Click here to verify your account

เมื่อเหยื่อทำการคลิกที่การเชื่อมโยงดังกล่าว ระบบก็จะทำการส่งค่าคุกกี้จากเครื่องคอมพิวเตอร์ผู้ใช้งานที่มีจุดอ่อนครอสไซต์สคริปติ้งไปยังเครื่องบริการของผู้โจมตี

ในความเป็นจริงแล้วผู้โจมตีอาจไม่จำเป็นต้องใช้การส่งอีเมลไปให้เหยื่อ แล้วรอให้เหยื่อทำการคลิกการเชื่อมโยงก็ได้ เพราะการโจมตีรูปแบบนี้ผู้โจมตีสามารถประยุกต์ใช้ส่วนย่อย (Element) ที่มีชื่อว่า <iframe> มาช่วยในการโจมตีได้ เพียงแค่เหยื่อทำการเปิดอ่านอีเมลแล้วบทคำสั่งก็จะทำงานโดยอัตโนมัติ

3) ครอสไซต์สคริปติ้งแบบชนิด 0 (DOM-based Cross-Site Scripting)

ครอสไซต์สคริปติ้งแบบชนิด 0 (DOM-based Cross-Site Scripting, Local cross-site scripting) หรือเรียกว่า “Type-0 Cross-Site Scripting” เป็นเทคนิคการโจมตีที่ไม่ใช่แบบถาวรและแบบชั่วคราว โดยบทคำสั่งจะไม่ถูกเก็บไว้บทเครื่องบริการหรือไม่มีการส่งบทคำสั่งไปยังเครื่องบริการ แต่เป็นการโจมตีโดยอาศัยคุณสมบัติของ Document Object Model (DOM) มาช่วยในการทำการโจมตี โดยนำข้อมูลที่รับเข้ามาแปลงให้เป็นบทคำสั่ง เช่น การที่เครื่องบริการไปอ่านข้อมูลจาก RSS feeds มาแล้วทำการแสดงผลไปยังเบราว์เซอร์โดยที่ไม่ทำการตรวจสอบก่อน

ผู้โจมตีที่จะใช้เทคนิคครอสไซต์สคริปติ้งแบบชนิด 0 จะต้องเข้าใจหลักการทำงานของการแสดงผลในการเตรียมเนื้อหา HTML ของเครื่องบริการเป็นอย่างดี เพราะสาเหตุหลักในการเกิดครอสไซต์สคริปติ้งชนิดนี้เกิดจากการไม่ได้ตรวจสอบการแสดงผลที่ครอบคลุมของเครื่องบริการ ทำให้ผู้โจมตีสามารถนำข้อมูลที่รับเข้ามาแปลงให้เป็นบทคำสั่งและทำให้เกิดการโจมตีชนิดนี้ได้

2.1.2 กฎการป้องกันครอสไซต์สคริปติ้ง (Cross-site scripting prevention rules) [6]

กฎทั้ง 8 ข้อ ถูกตั้งขึ้นมาเพื่อช่วยป้องกันปัญหาครอสไซต์สคริปติ้งส่วนใหญ่ในเว็บไซต์ระบบประยุกต์ โดยที่กฎส่วนใหญ่จะไม่อนุญาตสิทธิต่างๆ สำหรับข้อมูลที่ไม่น่าเชื่อถือในเอกสารเอชทีเอ็มแอล ซึ่งครอบคลุมกรณีส่วนใหญ่ของปัญหาครอสไซต์สคริปติ้งที่พบบ่อย

อาจไม่ต้องใช้กฎทั้งหมดในเว็บไซต์ระบบประยุกต์เพื่อป้องกันก็ได้ และกฎเหล่านี้สามารถเปลี่ยนแปลงตามการพัฒนาของเบราว์เซอร์ในอนาคตได้

1) RULE #0 - Never Insert Untrusted Data Except in Allowed Locations

กฎข้อ 0 เป็นวิธีการป้องกันในภาพรวม โดยไม่ใส่ข้อมูลที่ไม่น่าเชื่อถือลงในเอกสารเอชทีเอ็มแอล จนกว่าข้อมูลนั้นจะผ่านกฎข้อที่ 1 ถึงกฎข้อที่ 5 กฎข้อนี้มีขึ้นเพื่อป้องกันบริบทที่ผิดปกติจำนวนมากในเอกสารเอชทีเอ็มแอล หากต้องการจะใส่ข้อมูลที่ไม่น่าเชื่อถือลงในบริบทต้องทำการทดสอบในหลายๆ เบราว์เซอร์ก่อน

ที่สำคัญที่สุด ต้องไม่ยอมใช้งานคำสั่งจาวาสคริป (JavaScript code) จากแหล่งที่ไม่รู้จักหรือไม่น่าเชื่อถือ ตามที่แสดงด้านล่าง

<script>...NEVER PUT UNTRUSTED DATA HERE...</script>	*directly in a script*
<!--...NEVER PUT UNTRUSTED DATA HERE...-->	*inside an HTML comment*
<div ...NEVER PUT UNTRUSTED DATA HERE...=test />	*in an attribute name*
<NEVER PUT UNTRUSTED DATA HERE... href="/test" />	*in a tag name*
<style>...NEVER PUT UNTRUSTED DATA HERE...</style>	*directly in CSS*

2) RULE #1 - HTML Escape Before Inserting Untrusted Data into HTML Element Content

กฎข้อที่ 1 หลีกเลี่ยงการใส่ข้อมูลที่ไม่น่าเชื่อถือลงในเนื้อหาอีลีเมนท์ (Element) โดยเว็บไซต์ส่วนใหญ่จะหลีกเลี่ยงการใส่อักขระด้านล่างลงในเนื้อหาเอกสารเอชทีเอ็มแอลหรือแท็ก (Tag) ปกติภายใน เช่น div, p, b, td เป็นต้น ตามที่แสดงด้านล่าง

<body>...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...</body>
* ...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE... *
any other normal HTML elements

หลีกเลี่ยงการใช้อักขระดังต่อไปนี้โดยตรง เพื่อป้องกันการไม่ให้เป็นส่วนหนึ่งของคำสั่งจาวาสคริปโดยใช้การเข้ารหัสแบบ Hex ซึ่งจะแนะนำ 5 ตัวหลักที่มีผลกับ XML ตามที่แสดงด้านล่าง

```
& --> &amp;
< --> &lt;
> --> &gt;
" --> &quot;
' --> &#x27;     &apos; is not recommended
/ --> &#x2F;     forward slash is included as it helps end an HTML entity
```

3) RULE #2 - Attribute Escape Before Inserting Untrusted Data into HTML Common Attributes

กฎข้อที่ 2 หลีกเลี่ยงการใส่ข้อมูลที่ไม่น่าเชื่อถือในแอทริบิวต์ทั่วไปของเอชทีเอ็มแอล เช่น width, name, value เป็นต้น และแอทริบิวต์ที่มีความซับซ้อน เช่น href, src, style รวมไปถึงตัวจัดการเหตุการณ์ (Event handlers) เช่น onmouseover ตามที่แสดงด้านล่าง

```
<div attr=...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...>content</div>   inside UNquoted attribute
<div attr='...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...'>content</div> inside single quoted attribute
<div attr="...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...">content</div> inside double quoted
```

4) RULE #3 - JavaScript Escape Before Inserting Untrusted Data into JavaScript Data Values

กฎข้อที่ 3 หลีกเลี่ยงการใช้งานจาวาสคริปที่ไม่น่าเชื่อถือในส่วนของการใส่ข้อมูลจาวาสคริป หรือ ใช้งานจาวาสคริปด้วยความระมัดระวัง เพราะจาวาสคริปที่ไม่น่าเชื่อถือเป็นอันตรายและง่ายมากที่จะโดนโจมตีจากจุดนี้ ตามที่แสดงด้านล่าง

```
<script>alert('...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...')</script>
                                                                    inside a quoted string
<script>x='...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...'</script>
                                                                  one side of a quoted expression
<div onmouseover="x='...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...'"</div>
                                                                  inside quoted event handler
```

นอกจากนี้ยังมีจาวาสคริปบางอันไม่สามารถใช้งานได้อย่างปลอดภัยจากข้อมูลที่ไม่น่าเชื่อถือ ดังตัวอย่างด้านล่าง

```
<script>
        window.setInterval('...EVEN IF YOU ESCAPE UNTRUSTED DATA YOU ARE XSSED HERE...');
</script>
```

5) RULE #4 - CSS Escape And Strictly Validate Before Inserting Untrusted Data into HTML Style Property Values

กฎข้อที่ 4 หลีกเลี่ยง CSS และเคร่งครัดการตรวจค่าก่อนที่จะใส่ข้อมูลที่ไม่น่าเชื่อถือลงในค่าคุณสมบัติลักษณะเอชทีเอ็มแอล

```
<style>selector { property : ...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...; } </style>     property value
<style>selector { property : "...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE..."; } </style> property value
<span style="property : ...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...">text</style      property value
```

นอกจากนี้ยังมี CSS บางอันไม่สามารถใช้งานได้อย่างปลอดภัย จากข้อมูลที่ไม่น่าเชื่อถือ ดังตัวอย่างด้านล่าง

```
{ background-url : "javascript:alert(1)"; }  // and all other URLs
{ text-size: "expression(alert('XSS'))"; }   // only in IE
```

6) RULE #5 - URL Escape Before Inserting Untrusted Data into HTML URL Parameter Values

กฎข้อที่ 5 หลีกเลี่ยง URL ที่เป็นข้อมูลไม่น่าเชื่อถือก่อนนำใส่ลงไปในค่าพารามิเตอร์เอชทีเอ็มแอล ดังตัวอย่างด้านล่าง

```
<a href="http://www.somesite.com?test=...ESCAPE UNTRUSTED DATA BEFORE PUTTING HERE...">
link</a >
```

ถ้าใส่ข้อมูลที่ไม่น่าเชื่อถือลงไปใน href, src หรือคุณลักษณะที่อ้างอิงถึง URL อื่นๆ ควรจะทำการตรวจสอบให้แน่ใจว่ามันไม่ได้ชี้ไปที่ URL หรือจาวาสคริปอื่นที่เป็นอันตราย นอกจากนี้ URL นั้นควรจะถูกเข้ารหัสเพื่อความปลอดภัยโดยขึ้นอยู่กับบริบทของการแสดงผลข้อมูล ดังตัวอย่างด้านล่าง

```
String userURL = request.getParameter( "userURL" )
 boolean isValidURL = ESAPI.validator().isValidInput("URLContext", userURL, "URL", 255, false);
 if (isValidURL) {
        <a href="<%=encoder.encodeForHTMLAttribute(userURL)%>">link</a>
 }
```

2.1.2.7 RULE #6 - Use an HTML Policy engine to validate or clean user-driven HTML in an outbound way

OWASP AntiSamy

```
import org.owasp.validator.html.*;
Policy policy = Policy.getInstance(POLICY_FILE_LOCATION);
AntiSamy as = new AntiSamy();
CleanResults cr = as.scan(dirtyInput, policy);
MyUserDAO.storeUserProfile(cr.getCleanHTML()); // some custom function
```

OWASP Java HTML Sanitizer

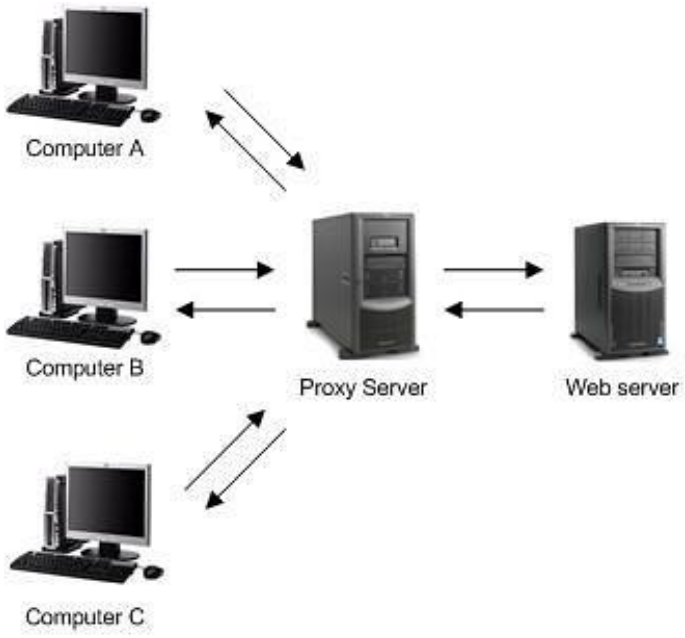
```
import org.owasp.html.Sanitizers;
import org.owasp.html.PolicyFactory;
PolicyFactory sanitizer = Sanitizers.FORMATTING.and(Sanitizers.BLOCKS);
String cleanResults = sanitizer.sanitize("<p>Hello, <b>World!</b>");
```

2.1.2.8 RULE #7 - Prevent DOM-based cross-site scripting

กฎข้อนี้ซับซ้อนและมีรายละเอียดเป็นพิเศษ ซึ่งส่วนใหญ่จะทำการป้องกันเกี่ยวกับคำสั่งจาวาสคริปเป็นหลัก

2.1.3 เครื่องบริการพร็อกซี่ (Proxy server)

เครื่องบริการพร็อกซี่ (Proxy server) [7] หรือที่นิยมเรียกกันสั้นๆ ว่าพร็อกซี่ (Proxy) คือเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นตัวกลางระหว่างเครื่องคอมพิวเตอร์ผู้ใช้งาน (Client computer) กับเครื่องบริการ (Server) จริง โดยเมื่อเราเปิดเว็บไซต์ผ่านระบบอินเทอร์เน็ตที่เครื่องคอมพิวเตอร์ผู้ใช้งาน ข้อมูลที่เราทำการส่งคำร้องขอ (Request) จะวิ่งไปที่พร็อกซี่ก่อน หลังจากนั้นคำร้องขอจากพร็อกซี่จะถูกส่งไปเครื่องบริการจริงต่อไป โดยที่คำร้องขอจะไม่ได้ถูกส่งจากเครื่องคอมพิวเตอร์ผู้ใช้งานไปยังเครื่องบริการตามปกติทั่วไป



รูปที่ 2 การทำงานของเครื่องบริการพร็อกซี่

เครื่องบริการพร็อกซี่มีประโยชน์ในการใช้งานหลายประการ โดยสามารถแบ่งจุดประสงค์หลักในการใช้งานได้ 3 อย่างคือ

1) เพื่อเพิ่มความเร็วในการเรียกดูหน้าเว็บไซต์จากอินเทอร์เน็ตได้ เนื่องจากเครื่องบริการพร็อกซี่จะทำการเก็บข้อมูลหน้าเว็บไซต์ที่ถูกร้องขอไปยังเครื่องบริการจากเครื่องคอมพิวเตอร์ผู้ใช้งานที่ส่งคำร้องขอรายแรกไป ทำให้เครื่องคอมพิวเตอร์ผู้ใช้งานรายอื่นๆ ที่ต้องการเปิดดูเว็บไซต์หน้าเดียวกันจะได้ข้อมูลโดยไม่ต้องส่งคำร้องขอไปยังเครื่องบริการอีกครั้ง ทำให้ประหยัดทั้งเวลาและแบนด์วิดท์ (Bandwidth) ของเครือข่าย ทำให้องค์กรขนาดใหญ่ อาทิ ผู้ที่ให้บริการอินเทอร์เน็ต (Internet Service Provide) บริษัทขนาดใหญ่ มหาวิทยาลัย เป็นต้น นิยมใช้บริการเครื่องบริการพร็อกซี่

2) เพื่อให้เครื่องคอมพิวเตอร์ผู้ใช้งานซ่อนตัว โดยเครื่องบริการพร็อกซี่จะช่วยปิดบังที่อยู่ไอพี (IP Address) จริงของเครื่องคอมพิวเตอร์ผู้ใช้งาน ทำให้เครื่องบริการไม่สามารถทราบได้ว่าที่อยู่ไอพีที่แท้จริงของเครื่องคอมพิวเตอร์ผู้ใช้งานคืออะไร โดยมากจะใช้ประโยชน์ในเรื่องนี้ในการปิดบังที่อยู่ไอพีเพื่อความปลอดภัยของผู้ใช้งาน

3) เพื่อคัดกรอง โดยเครื่องบริการพร็อกซี่สามารถช่วยทำการคัดการการใช้งานตามคุณสมบัติต่างๆ อาทิ การจำกัดสิทธิในการเข้าถึงเว็บไซต์บางแห่งที่มีเนื้อหาไม่เหมาะสม หรือจำกัดสิทธิ์การใช้งานของผู้ใช้งานในการเข้าใช้อินเทอร์เน็ตได้

ประเภทของเครื่องบริการพร็อกซี่ [7]

1) เครื่องบริการพร็อกซี่แคช (Caching proxy server)

เครื่องบริการพร็อกซี่แคช (Caching proxy server) เป็นเครื่องบริการพร็อกซี่ที่ใช้ประโยชน์เพื่อเร่งความเร็วในการเข้าสู่หน้าเว็บไซต์ต่างๆ โดยเก็บข้อมูลจากคำร้องขอของเครื่องคอมพิวเตอร์ผู้ใช้งานก่อนหน้า โดยเครื่องบริการพร็อกซี่แคชจะเก็บคำร้องขอที่ถูกร้องขอบ่อย

2) เว็บพร็อกซี่ (Web proxy)

เว็บพร็อกซี่ (Web proxy) เป็นพร็อกซี่ที่จะสนใจเฉพาะการใช้งานเชื่อมโยงไปยังเวิลด์ไวด์เว็บ (WWW) การทำงานส่วนใหญ่ของเว็บพร็อกซี่คือ เพื่อเก็บเว็บแคช (Web cache) และเพื่อทำการคัดกรองสิ่งต่างๆ ภายใต้เงื่อนไขของการคัดกรอง

3) ฮอสไทล์พร็อกซี่ (Hostile proxy)

ฮอสไทล์พร็อกซี่ (Hostile proxy) เป็นพร็อกซี่ที่ถูกนำไปใช้ในจุดประสงค์ที่ไม่ดี เพื่อดักเก็บข้อมูลที่ส่งกันระหว่างเครื่องคอมพิวเตอร์ผู้ใช้งานกับเครื่องบริการ อาทิ เก็บข้อมูลจากแบบฟอร์มต่างๆ รหัสผ่าน ฯ แต่ทั้งนี้สามารถใช้งาน SSL (Secure Sockets Layer) เพื่อความปลอดภัยจากการถูกดักเก็บได้

2.1.4 ModSecurity [8]

ModSecurity เป็นหน่วยงานที่ให้บริการไฟร์วอลล์ของเว็บระบบประยุกต์เสรี นอกจากนี้ยังทำหน้าที่ให้ข้อมูลเกี่ยวกับกฎความปลอดภัยด้านต่างๆ ของเว็บระบบประยุกต์ ซึ่งในกฎความปลอดภัยของ ModSecurity จะมีจุดเด่นอยู่ที่มีการถ่วงคะแนนด้วยค่าต่างๆ เพื่อให้คะแนนความความปลอดภัยแก่เว็บระบบประยุกต์ที่ต้องการนำไฟร์วอลล์เสรีดังกล่าวไปใช้งาน

ค่าดัชนีที่น่าสนใจในกฎความปลอดภัยของ ModSecurity มีหลายอย่าง อาทิ

1) ดัชนีของความสมบูรณ์ (maturity index) ซึ่งจะมีค่าดัชนีระหว่าง 0 - 9 โดย 0 หมายถึง เพิ่งทดลอง / เริ่มทดสอบ (beta / experimental) และ 9 หมายถึง ผ่านการทดสอบมาอย่างหนักแล้ว (heavily tested)

2) ดัชนีของความแม่นยำ (accuracy index) มีค่าดัชนีระหว่าง 0 - 9 โดย 0 หมายถึง ผิดพลาดสูง และ 9 หมายถึง ไม่มีรายงานของความผิดพลาด

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องมีดังต่อไปนี้

2.2.1 Defending against Cross-Site Scripting Attacks [9]

ชาร์และตันได้นำเสนอวิธีป้องกันครอสไซต์สคริปติ้ง 4 ประเภท คือ การแก้ปัญหาการเขียนคำสั่ง (defensive coding practices), การทดสอบครอสไซต์สคริปติ้ง (XSS testing), การตรวจจับช่องโหว่ (vulnerability detection), และการป้องกันระหว่างการใช้งานจริง (runtime attack prevention)

2.2.2 Semi-Automated XSS test using Firefox add-on [10]

เติมพรเลิศ และ ภิรมย์โสภา ได้นำเสนอวิธีการทดสอบครอสไซต์สคริปติ้งกึ่งอัตโนมัติด้วยไฟร์ฟอกซ์แอดออน โดยตรวจสอบจากสิ่งที่ข้อมูลควรจะถูกกรอง โดยอาศัยความรู้จากแหล่งความปลอดภัยต่างๆ ในการพิจารณา อีลีเมนต์ที่เสี่ยง แอททริบิวต์ที่เสี่ยง และคำสำคัญ

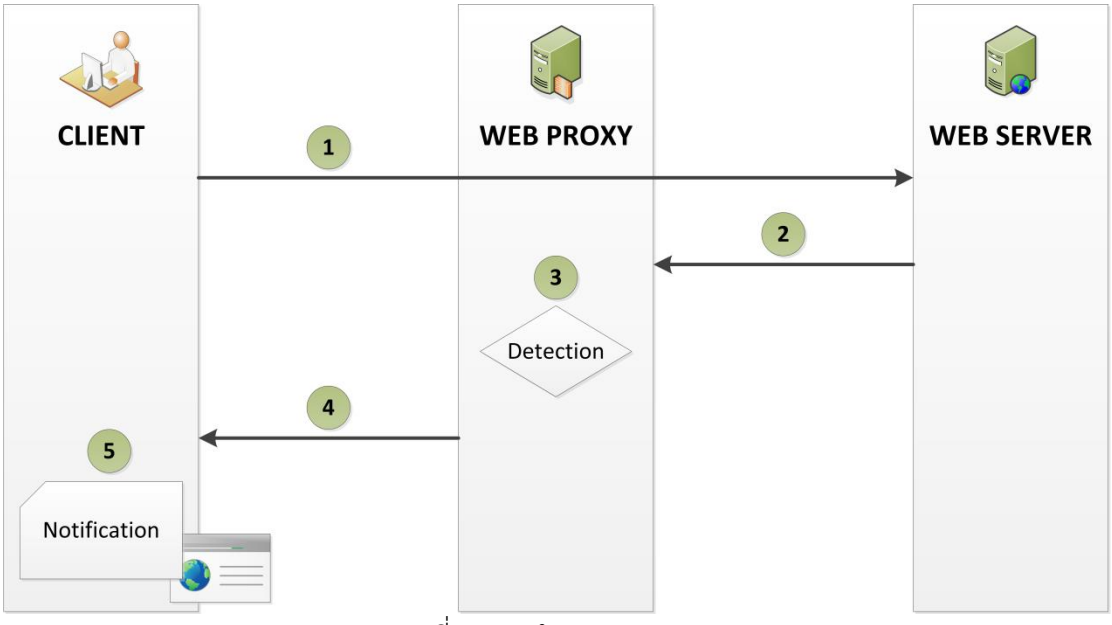
นอกจากนี้ยังนำเสนออีลีเมนต์ที่เสี่ยง แอททริบิวต์ที่เสี่ยงและคำสำคัญที่ไม่ได้ป้องกันออกมาให้ทราบด้วย ทั้งนี้โครงงานวิจัยนี้ได้ทดสอบวิธีการดังกล่าวกับแนวทางป้องกันแบบต่างๆ

บทที่ 3
การออกแบบระบบ

ในบทนี้จะกล่าวถึงการออกแบบระบบประยุกต์สำหรับตรวจจับปัญหาครอสไซต์สคริปติ้งในเว็บพร็อกซี่ โดยเนื้อหาในบทนี้จะอธิบายถึงภาพรวมของระบบ รวมไปถึงการทำงานของระบบในแต่ละขั้นตอน

3.1 ภาพรวมของระบบ

งานวิจัยนี้เน้นที่การทำงานของระบบประยุกต์สำหรับตรวจจับปัญหาครอสไซต์สคริปติ้งในเว็บพร็อกซี่ ซึ่งภาพรวมของระบบ คือ การติดตั้งเว็บพร็อกซี่ไว้ระหว่างกลางของการส่งข้อมูลเอกสารเอชทีเอ็มแอลระหว่างคอมพิวเตอร์ผู้ใช้งานและเครื่องที่ให้บริการ โดยการทำงานของระบบทั้งหมดจะเป็นไปตามแผนภาพด้านล่าง



รูปที่ 3 การทำงานของระบบ

3.2 การทำงานของระบบ

จากรูปที่ 3 จะเห็นภาพของการทำงานระบบทั้งหมด โดยจะถูกแบ่งออกเป็น 5 ขั้นตอนย่อยดังต่อไปนี้

1) คอมพิวเตอร์ผู้ใช้งานจะทำการส่งคำร้องขอไปยังเครื่องที่ให้บริการผ่านทางเว็บพร็อกซี่แต่เว็บพร็อกซี่จะยังไม่กระทำการตรวจจับใดๆ ในขั้นตอนนี้

2) เครื่องบริการทำการสร้างเอกสารเอชทีเอ็มแอลของเว็บระบบประยุกต์ขึ้นมา ตามคำร้องที่ได้รับมา และทำการส่งกลับไปยังคอมพิวเตอร์ผู้ใช้งานผ่านทางเว็บพร็อกซี่

3) ระบบประยุกต์ที่อยู่ในเว็บพร็อกซี่จะทำการตรวจประเมินเอกสารเอชทีเอ็มแอลที่ได้มา ว่ามีความเสี่ยงของปัญหาครอสไซต์สคริปติ้งหรือไม่ โดยให้ r เป็นร้อยละของความเสี่ยงปัญหาครอสไซต์สคริปติ้ง การคำนวณจะเป็นไปตามสมการที่ 1 ดังนี้

$$r = \sum_{i=1}^{I} r_i \times \frac{100}{I} \qquad (1)$$

โดยจะให้ r_i เป็นความเสี่ยงของปัญหาครอสไซต์สคริปติ้งที่ตำแหน่ง i ใดๆ ที่มีโอกาสเกิดปัญหาครอสไซต์สคริปติ้งซึ่งอยู่ในเอกสารเอชทีเอ็มแอล โดย i จะต้องอยู่ในเซ็ทของ { 1 , 2 , ... , i , ... , I-1 , I }

คะแนนสำหรับความเสี่ยงของปัญหาครอสไซต์สคริปติ้ง (r_i) นั้น จะอยู่ในช่วง 0.00 - 1.00 โดย 0.00 หมายถึง ไม่มีความเสี่ยงของปัญหาครอสไซต์สคริปติ้งในตำแหน่ง i นั้นๆ และ 1.00 หมายถึง ตำแหน่ง i นั้นๆ มีความเสี่ยงสูงสำหรับปัญหาครอสไซต์สคริปติ้ง ซึ่งการคำนวณเป็นไปตามสมการที่ 2

$$r_i = \frac{m_i + a_i}{M + A} \qquad (2)$$

ระบบประยุกต์ที่พัฒนาขึ้นได้นำการกำหนดค่ามาจากกฎของ ModSecurity ซึ่งให้ m_i เป็นดัชนีของความความสมบูรณ์ (maturity index) และให้ a_i เป็นดัชนีของความแม่นยำ (accuracy index) ณ ตำแหน่ง i ใดๆ โดยมี $M + A$ เป็นค่าสูงสุดของ $m_i + a_i$

4) หลังจากที่ระบบประยุกต์ในเว็บพร็อกซี่ได้ทำการตรวจจับเอกสารเอชทีเอ็มแอลแล้วพบว่าค่าร้อยละของ r มากกว่า 0 แสดงว่าเอกสารเอชทีเอ็มแอลดังกล่าวมีโอกาสเกิดปัญหาครอสไซต์สคริปติ้ง ถ้าหากค่าของ r ที่ปรากฏมีค่าสูง ก็แสดงว่าเว็บระบบ ป ร ะ ยุ ก ต์

ดังกล่าวมีความเสี่ยงของปัญหาครอสไซต์สคริปติ้งมาก เว็บพร็อกซี่ก็จะทำการส่งการแจ้งเตือนไปพร้อมกับเอกสารเอชทีเอ็มแอล

5) ถ้าผู้ใช้งานได้รับการแจ้งเตือนว่าเว็บระบบประยุกต์ที่จะเข้าเยี่ยมชมมีความเสี่ยง ผู้ใช้งานก็จะต้องทำการตัดสินใจว่าจะเชื่อหรือเพิกเฉยต่อการแจ้งเตือนที่ได้รับจากระบบประยุกต์ในเว็บพร็อกซี่

บทที่ 4
การสร้างและการทดลองระบบ

ในบทนี้จะเป็นการนำการออกแบบระบบจากบทที่แล้วมาทาให้เกิดผล โดยการนำแนวคิดดังกล่าวไปพัฒนาเป็นระบบประยุกต์ในเว็บพร็อกซี่ ซึ่งในบทนี้จะแบ่งออกเป็นสามส่วน ได้แก่ ส่วนแรกจะเป็นการอธิบายถึงแนวทางการทำให้เกิดผล ส่วนที่สองจะเป็นการอธิบายถึงเครื่องมือที่ใช้ในงานวิจัย ส่วนที่สามจะเป็นวิธีการประเมินวิจัย สรุป และอภิปรายผลการทดลองที่ได้

4.1 แนวทางการทำให้เกิดผล

งานวิจัยนี้ได้นำแนวทางการออกแบบระบบจากบทที่แล้ว มาทำการพัฒนาระบบประยุกต์ในเว็บพร็อกซี่ เพื่อทดลองตรวจจับปัญหาครอสไซต์สคริปติ้งบนเว็บระบบประยุกต์ ซึ่งแบ่งงานได้เป็นสามส่วนหลัก ได้แก่

4.1.1 ศึกษากฎจาก ModSecurity

นำกฎที่ได้มาจาก ModSecurity มาศึกษาโครงสร้างและรูปแบบของกฎ เพื่อดึงค่าและรูปแบบของกฎบางส่วนมาใช้งานในระบบประยุกต์ในเว็บพร็อกซี่ที่จะพัฒนาขึ้น ซึ่งในงานวิจัยนี้ได้นำค่าของดัชนีของความสมบูรณ์ (maturity index) และค่าของดัชนีของความแม่นยำ (accuracy index) มาใช้งานในการคำนวณผลร้อยละค่าความเสี่ยงของปัญหาครอสไซต์สคริปติ้ง

นอกจากนี้ ได้นำเอารูปแบบของกฎความปลอดภัยจาก ModSecurity ที่อยู่ในรูปแบบของนิพจน์ปรกติ (regular expression) มาปรับใช้งานในระบบประยุกต์เพื่อตรวจจับปัญหาครอสไซต์สคริปติ้งด้วย

4.1.2 พัฒนาระบบประยุกต์ในเว็บพร็อกซี่

ทำการพัฒนาระบบประยุกต์ในเว็บพร็อกซี่ที่จะใช้งานตรวจจับปัญหาครอสไซต์สคริปติ้ง ซึ่งในงานวิจัยนี้ได้เลือกใช้ภาษาจาวา (Java) ในการพัฒนาระบบประยุกต์ในเว็บพร็อกซี่ขึ้นมา

4.1.3 ทดลองตรวจจับปัญหาครอสไซต์สคริปติ้ง

หลังจากพัฒนาระบบประยุกต์ในเว็บพร็อกซี่จนสามารถใช้งานได้จริง ขั้นตอนต่อมาคือทำการทดลองตรวจจับปัญหาครอสไซต์สคริปติ้งบนเว็บระบบประยุกต์ โดยได้ทดลองสร้างเว็บไซต์ที่มีปัญหาครอสไซต์สคริปติ้งตามรูปแบบของกฎการป้องกันครอสไซต์สคริปติ้งเฉพาะกฎของที่ 1 – 5 กฎละจำนวน 10 เว็บไซต์ รวมทั้งหมด 50 เว็บไซต์ แล้วทำการทดลองโดยเรียกใช้งานเว็บไซต์ที่สร้างขึ้นทั้งหมดที่คอมพิวเตอร์ผู้ใช้งานไปยังเครื่องที่ให้บริการซึ่งมีเว็บพร็อกซี่ที่มีระบบประยุกต์ที่พัฒนาขึ้นอยู่ตรงกลาง

นอกจากนี้ก็ได้ทำการทดลองเชิงประสิทธิภาพ โดยทำการเปิดเว็บระบบประยุกต์ทั่วไปจำนวน 10 เว็บไซต์ซึ่งไม่ผ่านการใช้งานเว็บพร็อกซี่และเปิดผ่านการใช้งานเว็บพร็อกซี่เพื่อนำเวลาที่ใช้ในการสร้างเอกสารเอชทีเอ็มแอลทั้งหมดมาหาค่าเฉลี่ย และทำการเปรียบเทียบเชิงประสิทธิภาพกัน

4.2 เครื่องมือที่ใช้ในงานวิจัย

เครื่องมือที่ใช้ในงานวิจัยนี้ ประกอบด้วย

4.2.1 Eclipse

Eclipse เป็นโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java โดยข้อดีของโปรแกรม Eclipse คือ ติดตั้งได้ง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชั่น มี plugin ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF และที่สำคัญเป็น Freeware ใช้งานได้กับระบบปฏิบัติการ Windows, Linux และ Mac OS

4.2.2 Mozilla Firefox

Mozilla Firefox เป็นโปรแกรมเว็บบราวเซอร์โดยเป็นโปรแกรม Open Source ที่พัฒนาโดยองค์การที่ไม่หวังผลกำไรและบุคคลทั่วไป ใช้งานได้กับระบบปฏิบัติการ Windows, Linux และ Mac OS

4.2.3 LAMP

LAMP เป็นตัวอักษรย่อของโปรแกรม Open Source 4 ชนิดมารวมกัน เพื่อทำหน้าที่เป็นเครื่องบริการเว็บ (Web Server) อันประกอบด้วย Linux, Apache, MySQL และ PHP

- Linux เป็นระบบปฏิบัติการ (Operation System)
- Apache เป็นเครื่องบริการเว็บ (Web Server)
- MySQL เป็นฐานข้อมูล (Database)
- PHP เป็นภาษาคำสั่ง (Language)

4.2.4 Computer Laptop

Computer Laptop ที่ใช้งานในงานวิจัย เพื่อทดลองตรวจจับปัญหาครอสไซต์สคริปติ้ง มีรายละเอียดของเครื่อง ดังนี้

- Intel mobile core 2 duo t8100
- Memory DDR2 3072 MB
- เชื่อมต่อ internet download speed 10.82 Mbps

4.3 วิธีประเมินการวิจัย

การวิจัยนี้ได้กำหนดวิธีประเมินการวิจัยในสองแง่มุม ได้แก่

4.3.1 ผลเชิงคุณภาพ

งานวิจัยนี้ได้ทำการทดลองตรวจจับปัญหาของไซต์สคริปติ้งบนเว็บระบบประยุกต์ที่จัดทำขึ้น ตามรูปแบบของกฎการป้องกันครอสไซต์สคริปติ้ง เฉพาะกฎของที่ 1 – 5 กฎละจำนวน 10 เว็บไซต์ รวมทั้งหมด 50 เว็บไซต์

4.3.2 ผลเชิงประสิทธิภาพ

งานวิจัยนี้ได้ทำการเปิดเว็บระบบประยุกต์ทั่วไปจำนวน 10 เว็บไซต์ โดยเปิดไม่ผ่านการใช้งานเว็บพร็อกซี่ และเปิดผ่านการใช้งานเว็บพร็อกซี่ เพื่อนำเวลาที่ใช้ในการสร้างเอกสารเอชทีเอ็มแอลทั้งหมดมาหาค่าเฉลี่ย

4.4 สรุปผลการวิจัย

เนื่องจากการวิจัยนี้ได้ทดลองในสองแง่มุม และสรุปผลได้ดังนี้

4.4.1 ผลเชิงคุณภาพ

ระบบประยุกต์ในเว็บพร็อกซี่ที่ได้ทำการพัฒนาขึ้นมานั้น สามารถทำการตรวจจับปัญหาครอสไซต์สคริปติ้งบนเว็บระบบประยุกต์ที่สร้างมาเพื่อทดสอบตามรูปแบบของกฎการป้องกันครอสไซต์สคริปติ้ง เฉพาะกฎของที่ 1 – 5 กฎละจำนวน 10 เว็บไซต์ รวมทั้งหมด 50 เว็บไซต์ ได้ผลตามตารางที่ 3 ดังนี้

ตารางที่ 3 ตารางแสดงผลการทดลองตรวจจับปัญหาครอสไซต์สคริปติ้ง

Pattern	The number of attack	Runtime Detection
HTML Element Content	10	10
HTML Common Attributes	10	10
JavaScript Data	10	10
HTML Style Property	10	10
HTML URL Parameter	10	10

4.4.2 ผลเชิงประสิทธิภาพ

ผลการทดลองเปิดเว็บระบบประยุกต์ทั่วไป ให้สร้างเอกสารเอชทีเอ็มแอลจำนวน 10 เว็บไซต์ โดยเปิดไม่ผ่านการใช้งานเว็บพร็อกซี่ และเปิดผ่านการใช้งานเว็บพร็อกซี่ และนำเวลาที่ได้ทั้งหมดมาหาค่าเฉลี่ย ได้ผลตามตารางที่ 4 ดังนี้

ตารางที่ 4 ตารางแสดงผลการทดลองตรวจจับเปรียบเทียบเรื่องเวลาที่ใช้งาน

Samples	Avg. Time without Web Proxy	Avg. Time with Web Proxy
1	1,782 ms	3,008 ms
2	1,320 ms	2,886 ms
3	1,571 ms	3,310 ms
4	2,958 ms	4,625 ms
5	2,557 ms	4,001 ms
6	2,876 ms	4,513 ms
7	3,621 ms	5,792 ms
8	2,014 ms	3,979 ms
9	4,203 ms	7,818 ms
10	3,922 ms	7,380 ms
Average.	**2,682.40 ms**	**4,731.20 ms**

4.5 อภิปรายผลการวิจัย

จากผลการทดลองพบว่าระบบประยุกต์ในเว็บพร็อกซี่ที่พัฒนาขึ้นนั้น สามารถตรวจจับปัญหาครอสไซต์สคริปติ้งจากเว็บไซต์ตัวอย่างที่สร้างขึ้นจำนวน 50 เว็บไซต์ (ตามรูปแบบของกฎการป้องกันครอสไซต์สคริปติ้ง เฉพาะกฎของที่ 1 – 5 กฎละจำนวน 10 เว็บไซต์) ได้ร้อยละ 100

แต่การใช้งานเว็บพร็อกซี่ก็ใช้เวลาในการเพิ่มขึ้นโดยเฉลี่ย 2,048.80 มิลลิวินาที (เพิ่มขึ้นประมาณ 2 วินาที) หรือคิดเป็นร้อยละ 176.38 โดยเวลาที่เพิ่มขึ้นก็แลกเปลี่ยนกับความปลอดภัยที่เพิ่มขึ้นด้วย

บทที่ 5
บทสรุป

ในบทนี้จะแบ่งเนื้อหาออกเป็นสองส่วน ได้แก่ ส่วนแรกจะกล่าวถึงสิ่งที่ได้รับจากการวิจัย และส่วนที่สองจะกล่าวถึงแนวทางการวิจัยต่อไป

5.1 สิ่งที่ได้จากการวิจัย

สิ่งที่ได้จากการวิจัยนี้ ได้แก่

1) ได้ทราบถึงประเภทหลักของการปัญหาครอสไซต์สคริปติ้ง และกฎในการป้องกันปัญหาครอสไซต์สคริปติ้ง

2) ได้ทำการสำรวจและแบ่งรูปแบบของวิธีการป้องกันปัญหาครอสไซต์สคริปติ้งที่มีอยู่ในปัจจุบัน โดยนำเสนอหลักการ ทำการเปรียบเทียบข้อดีและข้อเสียของวิธีแต่ละรูปแบบสำหรับการป้องกันปัญหาดังกล่าว

3) ได้ทำการออกแบบและพัฒนาระบบประยุกต์ต้นแบบของเว็บพร็อกซี่ที่จะใช้งานเพื่อตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่ในขณะที่เว็บไซต์ใช้งานจริง ณ ฝั่งเครื่องคอมพิวเตอร์ผู้ใช้งานขึ้นมา และได้ทำการทดลองวัดผลของประสิทธิภาพเบื้องต้นในการทำงานจริง

4) ได้ทำการพัฒนาระบบประยุกต์ต้นแบบและทำการทดลองจริง และในอนาคตสามารถนำระบบประยุกต์นี้ไปพัฒนาต่อเพื่อใช้งานจริงต่อไปได้

5.2 แนวทางการวิจัยต่อ

งานวิจัยนี้ได้นำเสนอวิธีการตรวจจับปัญหาครอสไซต์สคริปติ้งโดยใช้เว็บพร็อกซี่ในขณะที่เว็บไซต์ใช้งานจริง ณ ฝั่งเครื่องคอมพิวเตอร์ผู้ใช้งานขึ้นมา โดยวิธีการนี้ได้แสดงให้เห็นว่าการตรวจจับปัญหาครอสไซต์สคริปติ้งนั้นสามารถทำได้จริง อย่างไรก็ตามยังมีแนวทางการวิจัยต่อที่สามารถแบ่งออกได้เป็น 3 ประเด็นดังนี้

1) ทำการพัฒนาระบบประยุกต์ในเว็บพร็อกซี่นี้ต่อ เพื่อนำไปใช้ตรวจปัญหาของความปลอดภัยอื่นๆ บนเว็บระบบประยุกต์

2) นำระบบประยุกต์นี้ไปทำการตรวจจับเว็บไซต์ต่างๆ ที่ต้องการทราบมาตรฐานจากความเสี่ยงในปัญหาครอสไซต์สคริปติ้ง

3) ปรับปรุงในจุดที่ยังบกพร่องของระบบประยุกต์ในเว็บพร็อกซี่ อาทิ วิธีการทำงานตรวจจับให้ได้ผลลัพธ์ที่เร็วยิ่งขึ้น, วิธีการคำนวณผลร้อยละของความเสี่ยงปัญหาครอสไซต์สคริปติ้ง, การแสดงผลให้ผู้ใช้งานเข้าใจได้ง่าย เป็นต้น

รายการอ้างอิง

[1] The Open Web Application Security Project. The Ten Most Critical Web Application Security Risks [Online]. 2013. Available from : http://owasptop10.googlecode.com/files/OWASP%20Top%2010%20-%202013%20-%20RC1.pdf [2013,April 20]

[2] Christey, S. ,and Martin, R.A. Vulnerability Type Distributions in CVE [Online]. 2007. Available from : https://www.cve.mitre.org/docs/vuln-trends/vuln-trends.pdf [2008,June 7]

[3] Wassermann, G. ,and Zhendong S. Static detection of cross-site scripting vulnerabilities. Software Engineering 2008 (May 2008) : 171-180.

[4] Johns, M., Engelmann, B.,and Posegga, J. XSSDS: Server-Side Detection of Cross-Site Scripting Attacks. Computer Security Applications Conference 2008 (December 2008) : 335-344.

[5] Wikipedia. Cross-site scripting [Online], Available from : http://en.wikipedia.org/wiki/Cross-site_scripting [2012,September 12]

[6] The Open Web Application Security Project. XSS (Cross Site Scripting) Prevention Cheat Sheet [Online]. 2012. Available from : http://www.owasp.com/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet [2012, September 12]

[7] Wikipedia. พร็อกซีเซิร์ฟเวอร์ [Online]. 2012. Available from : http://th.wikipedia.org/wiki/พร็อกซีเซิร์ฟเวอร์ [2012,September 12]

[8] ModSecurity. ModSecurity [Online]. 2012. Available from : http://www.modsecurity.org, [2013,April 20]

[9] Shar, L.K.,and Tan, H. K. Defending against Cross-Site Scripting Attacks Computer 45 (March 2012) : 55-62

[10] Termpornlerd, K.,and Piromsopa, K. Semi-Automated XSS test using Firefox add-on 12th National Computer Science and Engineering Conference (NCSEC2008) 2008 (November 2008)

ภาคผนวก

ตัวอย่าง : ชุดคำสั่งของเว็บพร็อกซี่ตรวจสอบเอกสารเอชทีเอ็มแอล

```
package scan;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class ScanHtml {

	public static String[] text ={"tempstring","qwertyuiopasdfghjklzxcvbnm"};
	public static String fileattacksName = "modsecurity_crs_41_xss_attacks.conf";
	public static String urlFile = "url.txt";
	public static String html = "html.txt";
	public static List<SecRule> secRuleList=new ArrayList<SecRule>();

	public static void setSecRule() {
		InputStream    fis;
		BufferedReader br;
		String         line;

		try {
```

```
                    fis = new FileInputStream(fileattacksName);

                    br = new BufferedReader(new InputStreamReader(fis,
Charset.forName("UTF-8")));
                    int index = 0;
                    while((line = br.readLine()) != null)
                    {
                        if(index==0)
                            line = line.substring(1);
                        if(line.indexOf("SecRule")==0){
                            secRuleList.add(new SecRule(line));
                        }
                        index++;
                    }
                } catch (FileNotFoundException e) {
                } catch (IOException e) {
                }
        }

        public static String addTextBody(String Html,String text){
            String newHtml="";
            int bodyBegin=0,bodyEnd=0;
            String lowerCase = Html.toLowerCase();
            bodyBegin = lowerCase.indexOf("<body");
            if(bodyBegin>=0)bodyEnd = lowerCase.indexOf(">",bodyBegin);
            if(bodyBegin>=0)
            {
                newHtml = Html.substring(0, bodyEnd+1) + text +
Html.substring(bodyEnd+1, Html.length()-1);
                return newHtml;
            }
```

```
        return Html;
}

public static int sacanTextCon(String Html,String text){
        int bodyBegin=0,bodyEnd=0,textindex=0;
        int a=0;
        String lowerCase = Html.toLowerCase();
        bodyBegin = lowerCase.indexOf("</head");
        if(bodyBegin>=0)bodyEnd = lowerCase.indexOf("</html",bodyBegin);
        if(bodyEnd>0)
        while(textindex<bodyEnd)
        {
                textindex = lowerCase.indexOf(text,bodyBegin);
                if(textindex ==-1) break;
                bodyBegin = textindex+1;
                if(bodyEnd>textindex) {
                        a++;
                }
        }
        return a;
}

public static int sacanSecRule(String Html,SecRule secRule){
        int countSecRule =0;
        String lowerCase = Html.toLowerCase();
        Pattern pattern = Pattern.compile ( secRule.operator );
        Matcher matcher = pattern.matcher ( lowerCase );

        while (matcher.find () )
        {
                countSecRule += matcher.groupCount();
```

```
                }

                return countSecRule;
        }

        public static int sacanTextConAttributeOnElement(String Html,String
element,String[] attribute){
                int bodyBegin=0,bodyEnd=0,textindex=0;
                int a=0;
                String lowerCase = Html.toLowerCase();
                bodyBegin = lowerCase.indexOf("</head");
                if(bodyBegin>=0)bodyEnd = lowerCase.indexOf("</html",bodyBegin);
                if(bodyEnd>0)
                while(textindex<bodyEnd)
                {
                        textindex = lowerCase.indexOf("<"+element,bodyBegin);
                        if(textindex ==-1) break;
                        int textEndindex = lowerCase.indexOf(">",textindex);
                        String elementText = lowerCase.substring(textindex,
textEndindex);
                        boolean hasAttribute = false;
                        for(int i = 0; i < attribute.length ; i++){
                                if(elementText.indexOf(attribute[i])>0) {
                                        hasAttribute =true;
                                        break;
                                }
                        }
                        bodyBegin = textindex+1;
                        if(bodyEnd>textindex&& hasAttribute) {
                                a++;
                        }
```

```
        }
        return a;
}

public static String sacanfile(String Html){
        String output = "";
        boolean scan = false;
        double sum = 0;
        double index = 0;
        int sumSacan = 0;
        Iterator<SecRule> iterator=secRuleList.iterator();
        int i = 0;
        while(iterator.hasNext())
        {
                i++;
                SecRule secRule=(SecRule)iterator.next();
                int sacan = sacanSecRule(Html,secRule);
                sumSacan += sacan;
                sum += sacan*(secRule.accuracy+secRule.maturity)/16;
                output += "secRule"+i+"="+sacan+"\n";
        }
        index = sum/sumSacan;
        output += "index=" + String.format("%1$,.2f", index*100);;
        if(index>0) scan = true;

        if(scan) output = "Yes & <b>Number of Risks</b>" + output;
        else output = "<b>Number of Risks</b>" + output;
        return output;
        }

public static boolean hasUrlFile(String url){
```

```
                InputStream     fis;
                BufferedReader br;
                String          line;

                try {
                        fis = new FileInputStream(urlFile);

                        br = new BufferedReader(new InputStreamReader(fis,
Charset.forName("UTF-8")));
                        int index = 0;
                        while((line = br.readLine()) != null)
                        {
                                if(index==0)
                                        line = line.substring(1);
                                if(line.equals(url)) return true;
                                index++;
                        }

                } catch (FileNotFoundException e) {
                } catch (IOException e) {
                }
                return false;
        }

        public static String newHtml(String url,String textOutput){
                String newHtml="";
                String newUrl = url;
                InputStream     fis;
                BufferedReader br;
                String          line;
```

```
		try {
			fis = new FileInputStream(html);

			br = new BufferedReader(new InputStreamReader(fis,
Charset.forName("UTF-8")));
			int index =0;
			while((line = br.readLine()) != null)
			{
				if(index==0)
					line = line.substring(1);
				newHtml += line;
				index++;
			}
			if(url.indexOf("?")>0) newUrl += "&proxyscan=1";
			else newUrl += "?proxyscan=1";
			newHtml = String.format(newHtml, textOutput,newUrl,url);
		} catch (FileNotFoundException e) {
		} catch (IOException e) {
		}
		//System.out.println(newHtml);
		return newHtml;
	}

}
```

ตัวอย่าง : เอกสารเอชทีเอ็มแอลที่นำมาใช้ทดสอบ

```
<html><head>
	<title> Guestbook </title>
	<meta name="Generator" content="EditPlus">
	<meta name="Author" content="">
	<meta name="Keywords" content="">
	<meta name="Description" content="">
</head>

<body>

	<form id="form1" name="form1" method="post" action="addguestbook.php">

		<table width="400" border="0" align="center" cellpadding="3"
cellspacing="1" bgcolor="#FFFFFF">
		<tbody><tr>
			<td width="117">Name</td>
			<td width="14">:</td>
			<td width="357"><input name="NAME" type="text" id="NAME"
size="40"></td>
		</tr>
		<tr>
			<td>E-mail</td>
			<td>:</td>
			<td><input name="EMAIL" type="text" id="EMAIL"
size="40"></td>
		</tr>
		<tr>
			<td valign="top">Comment</td>
			<td valign="top">:</td>
```

```
                    <td><textarea name="MESSAGE" cols="40" rows="3"
id="MESSAGE"></textarea></td>
                </tr>
                <tr>
                    <td> </td>
                    <td> </td>
                    <td><input type="submit" name="Submit" value="Submit">
<input type="reset" name="Submit2" value="Reset"></td>
                </tr>
                </tbody></table>

        </form>

        <br><br><br><br>

        <table width="400" border="0" align="center" cellpadding="3" cellspacing="0">
        <tbody><tr>
            <td><!--<strong>View Guestbook | <a href="guestbook.html">Sign
Guestbook</a> </strong>--></td>
        </tr>
        </tbody></table>
        <br>

        <table width="400" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
        <tbody><tr>
            <td><table width="400" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
                <tbody><tr>
                <td>ID</td>
                <td>:</td>
```

```
                <td>3</td>
            </tr>
            <tr>
                <td width="117">Name</td>
                <td width="14">:</td>
                <td width="357">attacker</td>
            </tr>
            <tr>
                <td>EMAIL</td>
                <td>:</td>
                <td></td>
            </tr>
            <tr>
                <td valign="top">Comment</td>
                <td valign="top">:</td>
                <td><script>document.location='http://attacker.abc/cookie-
steal.cgi?'+document.cookie</script></td>
            </tr>
            <tr>
                <td valign="top">Date/Time </td>
                <td valign="top">:</td>
                <td>2012-09-12 17:40:53</td>
            </tr>
            </tbody></table></td>
            </tr>
            </tbody></table>
            <br>
            <table width="400" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
            <tbody><tr>
```

```
                <td><table width="400" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
                <tbody><tr>
                <td>ID</td>
                <td>:</td>
                <td>2</td>
        </tr>
        <tr>
                <td width="117">Name</td>
                <td width="14">:</td>
                <td width="357">UBYI</td>
        </tr>
        <tr>
                <td>EMAIL</td>
                <td>:</td>
                <td>ubyiii@facebook.com</td>
        </tr>
        <tr>
                <td valign="top">Comment</td>
                <td valign="top">:</td>
                <td>test</td>
        </tr>
        <tr>
                <td valign="top">Date/Time </td>
                <td valign="top">:</td>
                <td>2012-09-11 18:43:58</td>
        </tr>
        </tbody></table></td>
        </tr>
        </tbody></table>
        <br>
```

```
<table width="400" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tbody><tr>
<td><table width="400" border="0" cellpadding="3" cellspacing="1"
bgcolor="#FFFFFF">
<tbody><tr>
<td>ID</td>
<td>:</td>
<td>1</td>
</tr>
<tr>
<td width="117">Name</td>
<td width="14">:</td>
<td width="357">yok</td>
</tr>
<tr>
<td>EMAIL</td>
<td>:</td>
<td></td>
</tr>
<tr>
<td valign="top">Comment</td>
<td valign="top">:</td>
<td>Hello world</td>
</tr>
<tr>
<td valign="top">Date/Time </td>
<td valign="top">:</td>
<td>2012-09-11 18:42:50</td>
</tr>
</tbody></table></td>
```

```
            </tr>
            </tbody></table>
            <br>

</body></html>
```

ประวัติผู้เขียนวิทยานิพนธ์

นายวรวิชญวิทย์ ประเสริฐยิ่ง เกิดเมื่อวันที่ 24 พฤษภาคม พ.ศ. 2529 ที่จังหวัดสุรินทร์ สำเร็จการศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการพัฒนาซอฟต์แวร์ จากจุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554