การทดสอบประสิทธิภาพของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันและแก้ไขช่องโหว่

นายอธิวัฒน์ สุภอมรพันธุ์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การทดสอบประสิทธิภาพของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันและแก้ไขช่องโหว่

นายอธิวัฒน์ สุภอมรพันธุ์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การทดสอบประสิทธิภาพของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันและแก้ไขช่องโหว่

โดย นายอธิวัฒน์ สุภอมรพันธุ์

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ

คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

.....

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เทอดพงษ์ แดงสี)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์)

..... กรรมการ

(รองศาสตราจารย์ ดร.พงษ์พิสิฐ วุฒิดิษฐโชติ)

ชื่อ : นายอธิวัฒน์ สุภอมรพันธุ์
ชื่อการค้นคว้าอิสระ : การทดสอบประสิทธิภาพของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันและแก้ไขช่องโหว่
สาขาวิชา : การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์
ปีการศึกษา : 2567

บทคัดย่อ

ในปัจจุบัน เทคโนโลยีปัญญาประดิษฐ์ (Artificial intelligence) โดยเฉพาะ Generative AI ได้ถูกนำมาใช้อย่างแพร่หลายในหลายด้าน ตั้งแต่การสร้างเนื้อหาอัตโนมัติ การพัฒนาแอปพลิเคชัน ไปจนถึงการช่วยเขียนและตรวจสอบโค้ดในอุตสาหกรรมซอฟต์แวร์ อย่างไรก็ตาม แม้ Generative AI จะมีศักยภาพสูง แต่ยังคงมีข้อผิดพลาดที่คาดเดาได้ยาก ซึ่งแตกต่างจากข้อผิดพลาดของมนุษย์และอาจส่งผลกระทบร้ายแรงหากไม่มีการจัดการที่เหมาะสม

การศึกษานี้มีวัตถุประสงค์เพื่อทดสอบความสามารถของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันที่มีความมั่นคงปลอดภัย รวมถึงประเมินศักยภาพของ AI ในการตรวจจับและแก้ไขช่องโหว่ของโค้ดที่สร้างขึ้นโดย AI เอง โดยการทดลองจะใช้ OWASP ZAP ซึ่งเป็นเครื่องมือมาตรฐานสำหรับการทดสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน เพื่อวิเคราะห์ช่องโหว่ของโค้ดที่สร้างโดย Generative AI ผลลัพธ์ที่ได้จะช่วยชี้ให้เห็นถึงขีดจำกัดของ AI ในด้านความมั่นคงปลอดภัยทางไซเบอร์ และแนวทางในการพัฒนา AI ให้สามารถสร้างซอฟต์แวร์ที่ปลอดภัยและเชื่อถือได้มากขึ้น

(มีจำนวนทั้งสิ้น 51 หน้า)

คำสำคัญ : Generative AI ความมั่นคงปลอดภัย การตรวจสอบโค้ด เว็บแอปพลิเคชัน OWASP ZAP ปัญญาประดิษฐ์

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Atiwat Supamornpan
Independent Study Title : Performance Testing of Generative AI in Developing Web Applications and Fixing Vulnerabilities
Major Field : Network and Information Security Management
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Nawaporn Wisitpongphan
Academic Year : 2024

ABSTRACT

In recent years, artificial intelligence (AI) technology, particularly Generative AI, has been widely applied across various fields, ranging from automatic content creation and application development to assisting in coding and code review in the software industry. Despite its high potential, Generative AI still produces unpredictable errors that differ from human mistakes and can lead to severe consequences if not managed properly.

This study aims to evaluate the capability of Generative AI in developing secure web applications and assess its potential in detecting and fixing vulnerabilities in the code it generates. The experiment will utilize OWASP ZAP, a standard tool for web application security testing, to analyze the vulnerabilities present in the code generated by Generative AI. The results obtained from this study will help identify the limitations of AI in cybersecurity and provide insights into how AI can be improved to create more secure and reliable software applications.

(Total 51 Pages)

Keywords: Generative AI, Software Security, Code Review, Web Applications, OWASP ZAP, Artificial Intelligence

_____ Advisor

กิตติกรรมประกาศ

การศึกษางานวิจัยอิสระในครั้งนี้ สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายฝ่าย ข้าพเจ้าขอแสดงความขอบคุณอย่างจริงใจต่อทุกท่านที่มีส่วนร่วมในการดำเนินงานวิจัยนี้

ขอขอบพระคุณ อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์ ซึ่งได้ให้คำแนะนำ องค์ความรู้ และข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่งตลอดระยะเวลาการดำเนินงานวิจัย คำชี้แนะจากท่านมีส่วนสำคัญอย่างยิ่งในการพัฒนาคุณภาพของงานวิจัยครั้งนี้

ขอขอบคุณ คณาจารย์และบุคลากรในสถาบันการศึกษา ที่ให้การสนับสนุนด้านข้อมูล คำปรึกษา ตลอดจนสิ่งอำนวยความสะดวกในการศึกษาวิจัย ทำให้สามารถดำเนินงานได้อย่างราบรื่น

ขอขอบคุณ เพื่อน ๆ นักศึกษา ที่ให้กำลังใจ คำแนะนำ และช่วยเหลือในด้านต่าง ๆ ไม่ว่าจะเป็นการแลกเปลี่ยนความคิดเห็น หรือการสนับสนุนด้านเทคนิค ซึ่งช่วยให้ข้าพเจ้ามีแรงผลักดันในการทำวิจัยจนสำเร็จ

สุดท้ายนี้ ขอขอบคุณ ครอบครัว ที่คอยเป็นกำลังใจและให้การสนับสนุนทั้งทางร่างกายและจิตใจมาโดยตลอด ความอดทนและความเข้าใจของท่านมีความหมายอย่างยิ่งต่อการดำเนินงานในครั้งนี้

อธิวัฒน์ สุภอมรพันธุ์

สารบัญ

หน้า

สารบัญตาราง

สารบัญรูปภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

มนุษย์มักทำผิดพลาดในชีวิตประจำวัน ซึ่งอาจส่งผลตั้งแต่เล็กน้อยจนถึงร้ายแรง เช่น การสูญเสียความไว้วางใจหรือความมั่นคงปลอดภัย เพื่อป้องกันข้อผิดพลาดเหล่านี้ มนุษย์ได้พัฒนากระบวนการต่าง ๆ เช่น การเปลี่ยนดีลเลอร์ในคาสิโน หรือการนับเครื่องมือในการผ่าตัด ในปัจจุบัน AI ถูกนำมาใช้งานแทนมนุษย์ในหลายด้าน เช่น Large Language Models (LLMs) แม้จะมีศักยภาพสูง แต่ข้อผิดพลาดของ AI กลับมีลักษณะแปลกและคาดเดาไม่ได้ แตกต่างจากข้อผิดพลาดของมนุษย์ จึงต้องมีการจัดการอย่างเหมาะสมเพื่อป้องกันผลกระทบที่อาจเกิดขึ้น [1]

ปัจจุบันมีการนำเทคโนโลยีเทคโนโลยีปัญญาประดิษฐ์ (AI) มาใช้งานกันอย่างแพร่หลาย ทั้งในองค์กร หรือแม้แต่ผู้ใช้งานทั่วไปก็สามารถเข้าถึงการใช้งานนี้ได้ ซึ่งเทคโนโลยีปัญญาประดิษฐ์ที่ใช้กันอย่างมากมายตอนนี้คือ Generative AI (Generative Artificial Intelligence) [2] คือเทคโนโลยีปัญญาประดิษฐ์ที่สามารถสร้างเนื้อหาใหม่ ๆ ขึ้นมาได้โดยใช้ข้อมูลที่มีอยู่เป็นพื้นฐาน ซึ่งเนื้อหาที่สร้างขึ้นอาจอยู่ในรูปของข้อความ รูปภาพ เสียง วิดีโอ หรือข้อมูลอื่น ๆ

การใช้งาน Generative AI สามารถพบได้ในหลากหลายด้าน เช่น การสร้างเนื้อหาข้อความ การเขียนบทความ ข่าว บทสนทนา หรือตอบคำถามอัตโนมัติ การสร้างภาพและวิดีโอ การสร้างภาพเหมือนจริงจากการอธิบายด้วยข้อความ การปรับปรุงคุณภาพภาพหรือสร้างวิดีโอจากข้อมูลเสียง การสร้างเสียงและดนตรี การสร้างเพลงหรือเสียงพูดที่มีความเป็นธรรมชาติ การออกแบบผลิตภัณฑ์: การสร้างตัวอย่างการออกแบบผลิตภัณฑ์ใหม่ ๆ หรือการปรับปรุงดีไซน์จากข้อมูลที่มีอยู่

รวมถึงในสายงานจำพวกด้าน Software Developer (ผู้พัฒนาซอฟต์แวร์) ที่สามารถนำ Generative AI มาช่วยในการทำงานได้อย่างหลากหลายเช่น สามารถนำ AI มาช่วยเขียนหรือแก้ไข Coding อีกทั้งยังสามารถใช้ให้ช่วยหาข้อผิดพลาดของโปรแกรมที่เขียนมาได้ด้วย แต่ด้วยจากบทความที่มีการตรวจสอบความถูกต้องของคำตอบที่ได้จากการถาม Generative AI แล้วไป

เทียบคำตอบกับทางเว็บไซต์อย่าง Stack overflow ซึ่งผลลัพธ์ที่ได้มาปรากฏว่าตัวคำตอบที่ได้มาจาก ChatGTP นั้นมีข้อมูลผิดพลาดกว่า 50% [3] ซึ่งเป็นตัวเลขที่ค่อนข้างสูงมาก

ทั้งนี้ผู้จัดทำจึงมีแนวคิดที่จะทำการวิจัยทดสอบความสามารถในการพัฒนาสร้างเว็บแอปพลิเคชัน [4] ที่มีความมั่นคงปลอดภัยจาก Generative AI รวมทั้งทดสอบความสามารถในการแก้ไขช่องโหว่ที่พบจาก Code ที่ได้มาจาก Generative AI

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อสร้างความตระหนักรู้ในการใช้ Generative AI ในการนำมาใช้เป็นเครื่องมือช่วยเขียนหรือพัฒนาโปรแกรม

1.2.2 เพื่อศึกษาความสามารถของ Generative AI ในการสร้างโปรแกรมที่มีความมั่นคงปลอดภัย และความสามารถในการแก้ไขช่องโหว่ที่พบ

1.2.3 เพื่อเปรียบเทียบเครื่องมือ Generative AI ในการสร้างเว็บแอปพลิเคชันที่มีความมั่นคงปลอดภัย

1.3 ขอบเขตของการวิจัย

1.3.1 ใช้ภาษาไทยเป็นหลัก ในการป้อนคำสั่งเข้าไปให้กับ Generative AI

1.3.2 เครื่องมือด้าน Generative AI ที่ใช้ในการศึกษา ประกอบด้วย

1.3.2.1 ChatGPT by OpenAI เวอร์ชัน ChatGPT-3.5

1.3.2.2 Gemini by Google เวอร์ชัน Gemini 1.5

1.3.2.3 Claude.ai เวอร์ชัน Claude 3.5 Sonnet

โดยทั้ง 3 เครื่องมือ Generative AI ผู้วิจัยใช้งานเวอร์ชันที่ไม่มีการเสียค่าใช้จ่ายใดๆ

1.3.3 ภาษาที่ใช้ในการพัฒนา

1.3.3.1 PHP ภาษาสคริปต์ (Scripting Language) ที่ออกแบบมาสำหรับการพัฒนาเว็บไซต์และการสร้างแอปพลิเคชันบนเซิร์ฟเวอร์ (Server-side scripting) โดยเฉพาะ

1.3.3.2 HTML (HyperText Markup Language) ใช้สำหรับสร้างและออกแบบโครงสร้างพื้นฐานของเว็บเพจ

1.3.3.3 CSS (Cascading Style Sheets) เป็นภาษาออกแบบ (Style Sheet Language) ที่ใช้สำหรับกำหนดลักษณะการแสดงผลของหน้าเว็บ เช่น สี ขนาดตัวอักษร ระยะห่าง หรือเลย์เอาต์ขององค์ประกอบใน HTML

1.3.4 ซอฟต์แวร์ที่ใช้

1.3.4.1 เครื่องมือทดสอบตรวจหาช่องโหว่ OWASP : ZAP เครื่องมือโอเพ่นซอร์สสำหรับการทดสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน โดยเป็นส่วนหนึ่งของโครงการ OWASP (Open Web Application Security Project)

1.3.4.2 เครื่องมือระบบฐานข้อมูล Database : XAMPP (mySQL) ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System: RDBMS) แบบโอเพ่นซอร์สที่ได้รับความนิยมอย่างแพร่หลาย ใช้สำหรับจัดการและจัดเก็บข้อมูลในรูปแบบโครงสร้างตาราง โดยสามารถใช้ภาษา SQL (Structured Query Language) ในการดำเนินการต่าง ๆ

1.3.4.3 เครื่องมือในการจำลอง Server PHP : XAMPP (Apache) Apache HTTP Server เป็นซอฟต์แวร์เว็บเซิร์ฟเวอร์แบบโอเพ่นซอร์สที่ใช้กันอย่างแพร่หลายที่สุดในโลก

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ทำให้ทราบศักยภาพของ Generative AI ในการพัฒนาหรือสร้างโปรแกรมอย่างมั่นคงปลอดภัย

1.4.2 ช่วยให้นักพัฒนาระบบตระหนักถึงขีดความสามารถของ Generative AI ในการสร้างเว็บแอปพลิเคชันที่มั่นคงปลอดภัย

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

งานวิจัยเรื่องการทดสอบความสามารถของ Generative AI ในการสร้างและแก้ไขช่องโหว่เว็บแอปพลิเคชัน นี้มุ่งเน้นในการวิเคราะห์และประเมินช่องโหว่ของเว็บแอปพลิเคชันที่ได้จากการสร้างมาจาก Generative AI เพื่อศึกษาความสามารถในการสร้างโปรแกรมอย่างมั่นคงและปลอดภัย รวมถึงความสามารถในการแก้ไขช่องโหว่ต่าง ๆ ที่พบ ผู้วิจัยจึงได้ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางการค้นคว้าดังต่อไปนี้

2.1 การประเมินช่องโหว่ของเว็บแอปพลิเคชัน

2.2 มาตรฐานความปลอดภัยของเว็บแอปพลิเคชัน

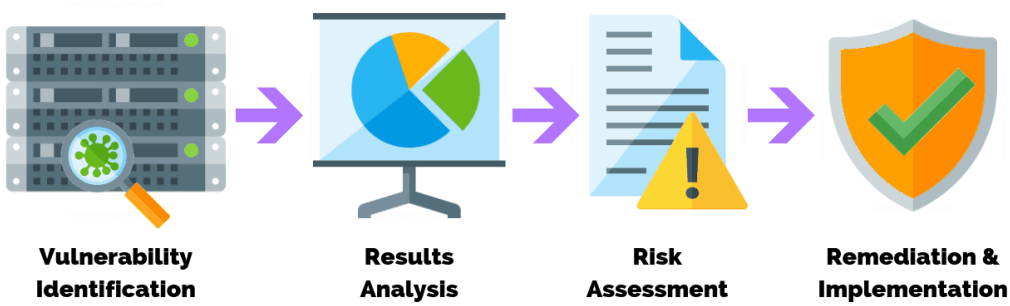
2.3 เครื่องมือสแกนช่องโหว่ของเว็บแอปพลิเคชัน

2.4 การประยุกต์ใช้งาน Generative AI ในปัจจุบัน

2.5 งานวิจัยที่เกี่ยวข้อง

2.1 การประเมินช่องโหว่ของเว็บแอปพลิเคชัน

การประเมินช่องโหว่ของเว็บแอปพลิเคชัน (Web Application Vulnerability Assessment) เป็นกระบวนการที่ใช้ในการตรวจสอบ วิเคราะห์ และประเมินความเสี่ยงที่อาจเกิดขึ้นจากช่องโหว่ภายในเว็บแอปพลิเคชัน ซึ่งช่องโหว่เหล่านี้อาจถูกผู้โจมตี (Hackers) ใช้ประโยชน์ในการเข้าถึงข้อมูลสำคัญ หรือก่อให้เกิดความเสียหายต่อระบบ การประเมินช่องโหว่นี้มีความสำคัญในการระบุและแก้ไขปัญหาก่อนที่การโจมตีจะเกิดขึ้นจริง โดยช่องโหว่ในเว็บแอปพลิเคชันอาจเกิดจากหลายสาเหตุ เช่น การพัฒนาระบบที่ขาดมาตรการรักษาความมั่นคงปลอดภัย การตั้งค่าระบบที่ไม่ถูกต้อง หรือการใช้ซอฟต์แวร์ที่ล้าสมัย ซึ่งอาจทำให้ระบบเสี่ยงต่อการถูกโจมตี



ภาพที่ 2-1 ขั้นตอนการประเมินช่องโหว่ (Vulnerability Assessment)

โดยขั้นตอนการประเมินช่องโหว่จะมี 4 ขั้นตอนดังภาพที่ 2-1

2.2.1 การระบุช่องโหว่ (Vulnerability Identification) ขั้นตอนนี้คือการค้นหาช่องโหว่ในเว็บแอปพลิเคชันโดยใช้เครื่องมือต่าง ๆ เช่น การสแกนช่องโหว่ การตรวจสอบโค้ด และการทดสอบเจาะระบบ เพื่อระบุจุดอ่อนที่อาจถูกโจมตี

2.2.2 การวิเคราะห์ช่องโหว่ (Results Analysis) การวิเคราะห์ช่องโหว่จะประเมินถึงวิธีที่ผู้โจมตีสามารถใช้ประโยชน์จากช่องโหว่และผลกระทบที่อาจเกิดขึ้น เพื่อจัดลำดับความร้ายแรงของช่องโหว่แต่ละตัว

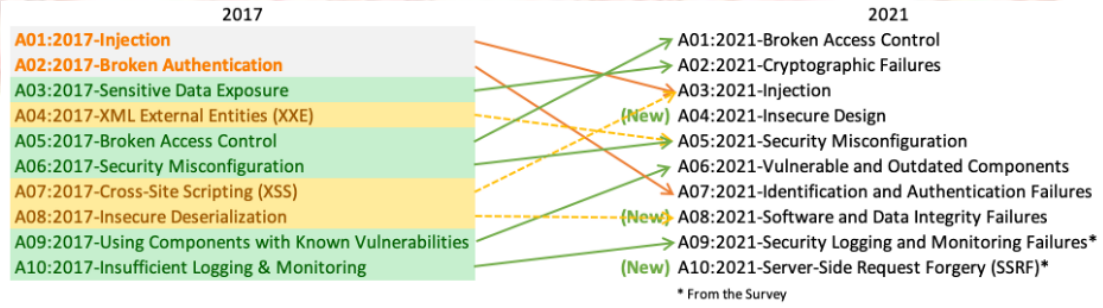
2.2.3 การประเมินความเสี่ยง (Risk Assessment) การประเมินความเสี่ยงคำนึงถึงความร้ายแรงและความน่าจะเป็นที่ช่องโหว่จะถูกโจมตี เพื่อกำหนดระดับความเสี่ยงและลำดับความสำคัญในการแก้ไขช่องโหว่

2.2.4 การแก้ไข (Remediation & Implementation) การแก้ไขช่องโหว่เป็นขั้นตอนสุดท้ายในกระบวนการประเมินช่องโหว่ ซึ่งเกี่ยวข้องกับการดำเนินการเพื่อแก้ไขหรือป้องกันไม่ให้ช่องโหว่ที่ระบุและประเมินความเสี่ยงแล้วกลายเป็นจุดอ่อนที่สามารถถูกโจมตีได้ การแก้ไขช่องโหว่อาจทำได้หลายวิธี เช่น การปรับปรุงโค้ดแหล่งที่มา (Source Code Improvements) การติดตั้งแพตช์ (Patch Installation) การแก้ไขช่องโหว่ที่เกิดขึ้นจะช่วยลดความเสี่ยงที่อาจส่งผลกระทบต่อความมั่นคงปลอดภัยของเว็บแอปพลิเคชันและข้อมูลของผู้ใช้งาน

2.2 มาตรฐานความปลอดภัยของเว็บแอปพลิเคชัน (Web Application Security Standards)

มาตรฐานความมั่นคงปลอดภัยของเว็บแอปพลิเคชันคือแนวทางหรือข้อกำหนดที่ใช้ในการออกแบบ พัฒนา และบริหารจัดการเว็บแอปพลิเคชัน เพื่อป้องกันภัยคุกคามทางไซเบอร์และลดความเสี่ยงจากการโจมตี มาตรฐานเหล่านี้ช่วยป้องกันการรั่วไหลของข้อมูลและสร้างความน่าเชื่อถือให้กับระบบ

หนึ่งในมาตรฐานที่ได้รับการยอมรับอย่างกว้างขวางคือ OWASP Application Security Verification Standard (ASVS) ซึ่งพัฒนาโดยโครงการ Open Web Application Security Project (OWASP) โดยเป็นแนวทางสำคัญในการประเมินและตรวจสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน มาตรฐานนี้กำหนดข้อกำหนดความมั่นคงปลอดภัยในหลายด้าน เช่น การยืนยันตัวตน (Authentication) มาตรการเพื่อป้องกันการเข้าถึงระบบโดยไม่ได้รับอนุญาต, การป้องกันข้อมูล (Data Protection) การรักษาความลับและความถูกต้องของข้อมูล หรือแม้แต่ การจัดการช่องโหว่ (Vulnerability Management) การตรวจสอบและแก้ไขช่องโหว่เพื่อป้องกันการโจมตี มาตรฐานดังกล่าวมีบทบาทสำคัญในการช่วยให้นักพัฒนาและองค์กรสามารถสร้างและดูแลระบบที่มีความมั่นคงปลอดภัยสูง รวมถึงปฏิบัติตามข้อกำหนดด้านความมั่นคงปลอดภัยที่เป็นสากลอย่างมีประสิทธิภาพ



ภาพที่ 2-2 OWASP Top 10 Web Application Security Risks ในปี 2021

จากภาพที่ 2-2 เป็นรายการความเสี่ยงด้านความมั่นคงปลอดภัยสำคัญที่สุดสำหรับเว็บแอปพลิเคชัน ซึ่งจัดทำโดยโครงการ OWASP โดยอัปเดตล่าสุดในปี 2021 เพื่อช่วยให้นักพัฒนาและองค์กรเข้าใจและป้องกันความเสี่ยงที่พบบ่อยที่สุด โดยสรุปได้ดังนี้ [5]

2.2.1 Broken Access Control การควบคุมการเข้าถึงที่ไม่เหมาะสม ทำให้ผู้โจมตีสามารถเข้าถึงข้อมูลหรือฟังก์ชันที่ไม่ได้รับอนุญาตได้

2.2.2 Cryptographic Failures การจัดการข้อมูลที่สำคัญ เช่น ข้อมูลส่วนบุคคลหรือรหัสผ่าน ไม่เป็นไปตามมาตรฐานการเข้ารหัสที่ปลอดภัย

2.2.3 Injection การโจมตีผ่านการใส่โค้ดที่เป็นอันตราย เช่น SQL Injection หรือ Command Injection เพื่อควบคุมระบบหรือดึงข้อมูลออกมา

2.2.4 Insecure Design การออกแบบระบบที่ขาดความรัดกุมด้านความมั่นคงปลอดภัย ทำให้ระบบเสี่ยงต่อการโจมตี

2.2.5 Security Misconfiguration การตั้งค่าความมั่นคงปลอดภัยที่ไม่เหมาะสม เช่น การเปิดใช้งานฟีเจอร์ที่ไม่จำเป็น หรือใช้การตั้งค่าดั้งเดิม (Default Settings)

2.2.6 Vulnerable and Outdated Components การใช้ซอฟต์แวร์หรือไลบรารีที่ล้าสมัยหรือมีช่องโหว่ ทำให้แอปพลิเคชันเสี่ยงต่อการถูกโจมตี

2.2.7 Identification and Authentication Failures ความล้มเหลวในการตรวจสอบตัวตนหรือการจัดการเซสชัน เช่น การตั้งค่ารหัสผ่านที่อ่อนแอ หรือการขโมยเซสชัน

2.2.8 Software and Data Integrity Failures ความล้มเหลวในการตรวจสอบความถูกต้องของซอฟต์แวร์หรือข้อมูล เช่น การโจมตี Supply Chain หรือการปรับแต่งโค้ดโดยไม่ได้รับอนุญาต

2.2.9 Security Logging and Monitoring Failures การขาดการบันทึกและติดตามเหตุการณ์ความมั่นคงปลอดภัย ทำให้ไม่สามารถตรวจจับและตอบสนองต่อการโจมตีได้ทันเวลา

2.2.10 Server-Side Request Forgery (SSRF) การโจมตีที่ทำให้เซิร์ฟเวอร์ส่งคำขอไปยังปลายทางที่ไม่ได้รับอนุญาต ซึ่งอาจนำไปสู่การเข้าถึงข้อมูลสำคัญ

2.3 เครื่องมือสแกนช่องโหว่ของเว็บแอปพลิเคชัน

เครื่องมือสแกนช่องโหว่ของเว็บแอปพลิเคชัน (Web Application Vulnerability Scanner) คือซอฟต์แวร์ที่พัฒนาขึ้นเพื่อระบุและตรวจสอบช่องโหว่ในระบบเว็บแอปพลิเคชันโดยอัตโนมัติ โดยมุ่งเน้นการค้นหาช่องโหว่ที่อาจถูกผู้โจมตีใช้ประโยชน์ เช่น SQL Injection, Cross-Site

Scripting (XSS) และ Cross-Site Request Forgery (CSRF) นอกจากนี้ เครื่องมือยังสร้างรายงานที่ระบุรายละเอียดช่องโหว่และแนวทางแก้ไขอย่างเป็นระบบ เพื่อช่วยให้องค์กรสามารถแก้ไขปัญหาได้อย่างมีประสิทธิภาพ ตัวอย่างเครื่องมือสแกนช่องโหว่ที่ได้รับความนิยม

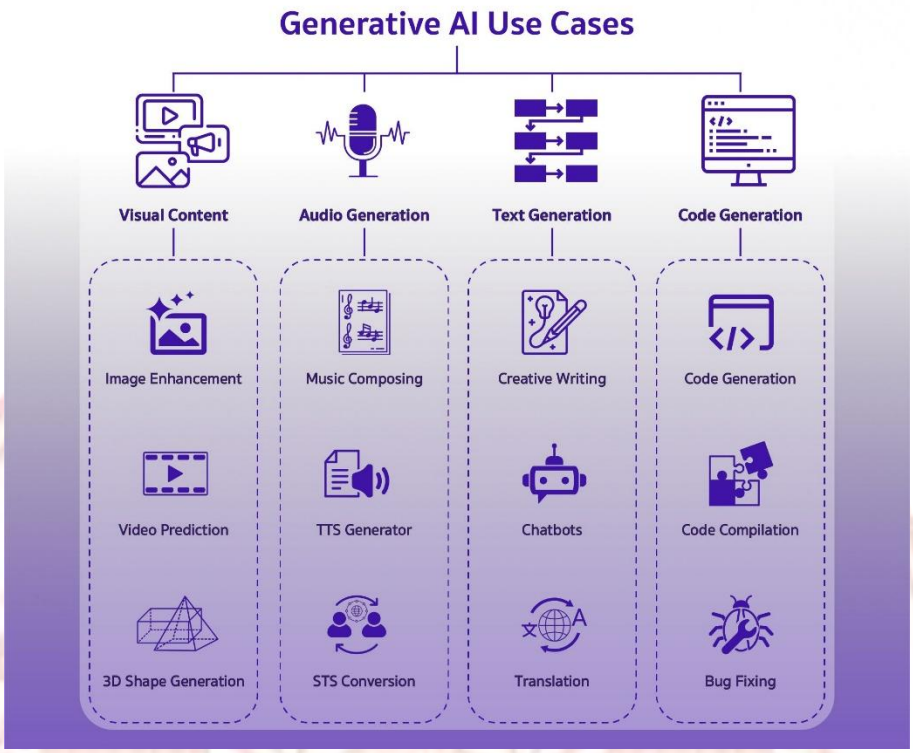
2.3.1 OWASP ZAP (Zed Attack Proxy) [6] เครื่องมือโอเพ่นซอร์สที่ใช้ตรวจสอบช่องโหว่ของเว็บแอปพลิเคชัน รองรับการสแกนทั้งแบบอัตโนมัติและด้วยตนเอง โดยเหมาะสำหรับนักพัฒนาและนักทดสอบระบบที่ต้องการเครื่องมือที่มีประสิทธิภาพและใช้งานฟรี

2.3.2 Burp Suite [7] เครื่องมือสำหรับการทดสอบเจาะระบบ (Penetration Testing) ซึ่งสามารถสแกนและวิเคราะห์ช่องโหว่ในเว็บแอปพลิเคชันได้อย่างลึกซึ้ง พร้อมทั้งมีฟีเจอร์ที่ช่วยนักพัฒนาแก้ไขปัญหาและปรับปรุงความมั่นคงปลอดภัยของระบบ

2.3.3 Nikto [8] เป็นเครื่องมือสแกนเว็บเซิร์ฟเวอร์แบบโอเพนซอร์สที่ใช้ทดสอบความมั่นคงปลอดภัยของเว็บเซิร์ฟเวอร์และเว็บแอปพลิเคชัน ทำงานโดยการทดสอบเว็บเซิร์ฟเวอร์เพื่อค้นหาไฟล์ที่อาจเป็นอันตราย, โปรแกรมหรือสคริปต์ที่มีความเสี่ยง รวมถึงตรวจสอบการตั้งค่าที่ไม่เหมาะสมและการอัพเดตที่ล้าสมัย

2.4 การประยุกต์ใช้งาน Generative AI ในปัจจุบัน

Generative AI เป็นเทคโนโลยีปัญญาประดิษฐ์ที่สามารถสร้างสรรค์ผลงานใหม่โดยอิงจากชุดข้อมูลที่ใช้ฝึกโมเดล การพัฒนา Generative AI ใช้เทคนิคการเรียนรู้ของเครื่อง ทั้งแบบมีผู้สอน (Supervised Learning) และไม่มีผู้สอน (Unsupervised Learning) โดยมีความสามารถครอบคลุมตั้งแต่การสร้างข้อความ รูปภาพ เสียง จนถึงการเขียนโค้ด ซึ่งช่วยเพิ่มประสิทธิภาพ ลดต้นทุน และประหยัดเวลาในการทำงาน การประยุกต์ใช้ Generative AI ในธุรกิจด้านวิศวกรรมซอฟต์แวร์ ตัวอย่างเช่น ช่วยเขียนโค้ด ติดตาม และแก้ไขข้อผิดพลาดในซอฟต์แวร์, พัฒนา Prototype และ UI อย่างรวดเร็ว หรือแม้แต่สนับสนุนการพัฒนาแอปพลิเคชันด้วยตัวอย่างโค้ดในหลากหลายภาษา



ภาพที่ 2-3 Generative AI User Cases

จากภาพที่ 2-3 จะเห็นได้ว่ายังมีอีกธุรกิจหลายอุตสาหกรรมที่จะได้ประโยชน์จากการใช้ Generative AI การใช้งานที่มีความหลากหลายของ Generative AI ทำให้เกิดนวัตกรรมใหม่ ๆ มากกว่าแค่การสร้างระบบอัตโนมัติ แต่ยังช่วยให้เราสามารถจัดการกับงานที่มีความซับซ้อนและเพิ่มประสิทธิภาพในการดำเนินงานได้อย่างรวดเร็ว [9]

ปัจจุบัน Generative AI ยังไม่สามารถสร้างข้อมูลที่ถูกต้องได้ร้อยเปอร์เซ็นต์ ซึ่งข้อมูลที่ไม่ถูกต้องที่ถูกสร้างขึ้นมาจากโมเดลภาษาขนาดใหญ่ หรือ Misinformation เหล่านี้อาจส่งผลกระทบอย่างรุนแรงต่อสังคม เนื่องจากสามารถทำให้ประชาชนเข้าใจผิดและส่งผลต่อการดำเนินชีวิตประจำวัน อาจก่อให้เกิดความวุ่นวายและทำลายกลไกการทำงานปกติของบุคคลหรือสังคม [10] โดยเฉพาะอย่างยิ่งเมื่อมีการนำ AI ไปใช้ในทางที่ผิดเพื่อหลอกลวงผู้คน [11]

สำหรับนักพัฒนาโปรแกรม เครื่องมือ AI สำหรับก็ถูกนำใช้อย่างแพร่หลายในการเขียนโปรแกรมแบบคู่ (AI Pair-Programming) เช่น GitHub Copilot [12] ซึ่งเครื่องมือนี้ได้เปลี่ยนพฤติกรรมของนักพัฒนาจากการ เขียนโค้ด ไปสู่การ ทำความเข้าใจโค้ด

อย่างไรก็ตาม นักพัฒนาก็ยังจำเป็นต้องทำการ แก้ไขข้อผิดพลาด และปรับแต่งโค้ด เพื่อให้สามารถนำโค้ดที่ได้ไปใช้ในบริบทของโปรแกรมของตนเองได้ ซึ่งอาจส่งผลกระทบต่อประสิทธิภาพในการทำงาน และด้วยอีกหนึ่งเหตุผลสำคัญคือ เครื่องมือ AI เหล่านี้ไม่ได้ให้คำตอบที่ถูกต้องเสมอไป และอาจสร้างข้อผิดพลาดหรือช่องโหว่ด้านความมั่นคงปลอดภัยในโค้ด หากไม่มีการตรวจสอบและแก้ไขโดยโปรแกรมเมอร์หรือผู้เชี่ยวชาญ [13]

ในช่วงเวลาที่ผู้วิจัยเริ่มดำเนินการวิจัยทดลองในหัวข้อนี้ คือช่วงเดือน ตุลาคม ปี พ.ศ. 2567 มีเครื่องมือ Generative AI ที่มีความสามารถโดดเด่นเกิดขึ้นมากมาย ผู้วิจัยจึงได้คัดเลือกเครื่องมือ AI จำนวน 3 ตัว ได้แก่ ChatGPT, Gemini และ Claude ซึ่งเป็นเครื่องมือที่ได้รับความนิยมและถูกกล่าวถึงอย่างแพร่หลายใน 10 อันดับแรก ของช่วงเวลานั้น [14]

2.5 งานวิจัยที่เกี่ยวข้อง

Samia Kabir และคณะ [15, 16] ได้ศึกษาวิเคราะห์การตอบคำถามของ ChatGPT ต่อคำถามใน Stack Overflow เพื่อประเมินความถูกต้องแม่นยำ ความสม่ำเสมอ ครบถ้วน และความกระชับของ ChatGPT พบว่า 52% ของคำตอบ ChatGPT มีข้อมูลผิดพลาดและ 77% ของคำตอบมีความยาวเกินจำเป็น และยังทำการศึกษาเกี่ยวกับผู้ใช้ สำรวจความคิดเห็นของผู้ใช้ที่เป็นโปรแกรมเมอร์ พบว่าผู้ใช้จำนวน 35% ยังคงชอบคำตอบของ ChatGPT เนื่องจากความครบถ้วนและรูปแบบการเขียนที่ดี และผู้ใช้ที่ยอมมองข้ามข้อมูลที่ผิดพลาดในคำตอบมีมากถึง 39% นอกจากนี้ยังมีการศึกษาเพิ่มเติมเกี่ยวกับความมั่นคงปลอดภัยของโค้ดที่ถูกสร้างโดย ChatGPT โดยทีมวิจัยได้ทำการทดลองให้ ChatGPT สร้างโปรแกรมคอมพิวเตอร์จำนวนหนึ่ง เพื่อตรวจสอบความมั่นคงปลอดภัยของโค้ดที่สร้างขึ้นมา โดยโปรแกรมเหล่านี้ถูกเขียนด้วยภาษาต่าง ๆ เช่น C, C++, Python, HTML และ Java งานวิจัยนี้ชี้ให้เห็นว่า แม้ ChatGPT จะสามารถสร้างโค้ดที่ดูเหมือนจะทำงานได้ แต่ก็ยังมีความเสี่ยงด้านความมั่นคงปลอดภัยอยู่ดี

อีกทั้งยังมีทีมวิจัยได้ทำการเลือกข้อผิดพลาดที่พบบ่อยในการเขียนโปรแกรม C/C++ ที่อยู่ในแพลตฟอร์ม Sifu มาทำ Cybersecurity Challenges กับ ChatGPT เวอร์ชั่น 13 มกราคม 2023 ซึ่งใช้ข้อมูลการฝึกจาก GPT-3 โค้ดที่มีช่องโหว่จะถูกนำเสนอให้ ChatGPT เพื่อทำการระบุและแก้ไขช่องโหว่ ทีมวิจัยได้ทำการทดสอบผ่านคำถามและคำตอบระหว่างมนุษย์กับ ChatGPT โดยคำถามถูกออกแบบให้ตรวจสอบว่า ChatGPT สามารถระบุปัญหาและแก้ไขโค้ดได้ถูกต้องหรือไม่ [17] ผู้วิจัยได้ทำการทดลองเพื่อประเมินประสิทธิภาพของ ChatGPT และพบว่ามีข้อดีหลายประการ แต่ก็ได้มีการระบุข้อจำกัดหลายประการ รวมถึงการขาดความเข้าใจ

บริบทของ ChatGPT ความเป็นไปได้ในการเปลี่ยนความหมายของโค้ด รวมไปถึงภาษาอื่น ๆ สำหรับทำเว็บแอปพลิเคชัน สรุปได้ว่า ChatGPT มีศักยภาพในการสร้างโค้ดที่ปลอดภัยสำหรับเว็บแอปพลิเคชัน แต่ต้องได้รับคำสั่งที่เฉพาะเจาะจงในการระบุช่องโหว่ที่ต้องการแก้ไข [18]

นอกจากนี้ยังมีการศึกษาเกี่ยวกับการเลือกใช้ภาษาที่ใช้สำหรับป้อนเข้าสู่ Generative AI โดย Rei Yamagishi และคณะ [19] ได้ศึกษาวิจัยถึงผลกระทบของการใช้ภาษาป้อนข้อมูลที่แตกต่างกัน (ภาษาญี่ปุ่นและภาษาอังกฤษ) ต่อความมั่นคงปลอดภัยของโค้ดที่สร้างโดย ChatGPT ผลการวิเคราะห์พบว่าโค้ดที่ไม่ปลอดภัยถูกสร้างขึ้นในทั้งสองภาษา แต่ส่วนใหญ่ไม่ขึ้นอยู่กับภาษาที่ป้อน ผู้ใช้ควรประเมินและปรับปรุงความมั่นคงปลอดภัยของโค้ดด้วยตนเอง เนื่องจากคำอธิบายที่แนบมากับโค้ดมักจะไม่เพียงพอในการสร้างโค้ดที่ปลอดภัย

การเลือกใช้งานเครื่องมือที่เหมาะสมสำหรับการนำมาตรวจสอบช่องโหว่ก็สำคัญไม่แพ้กัน ซึ่ง Larry Suto [20] ชี้ให้เห็นว่าเครื่องมือสแกนเนอร์เพื่อความมั่นคงปลอดภัยของเว็บแอปพลิเคชันยังคงมีช่องโหว่ที่พลาด แม้จะได้รับการฝึกฝน ดังนั้น ความแม่นยำยังควรเป็นจุดสำคัญที่ทีมความมั่นคงปลอดภัยควรให้ความสำคัญเมื่อเลือกใช้เครื่องมือสแกน [21, 22, 23]

จากการศึกษางานวิจัยที่เกี่ยวข้องข้างต้น พบว่างานวิจัยส่วนใหญ่ยัง มุ่งเน้นไปที่การนำ Generate AI มาวิเคราะห์ข้อผิดพลาดหรือช่องโหว่ทั่วไปของระบบ โดยไม่ได้ให้ความสำคัญกับความสามารถของ Generative AI ในการสร้างเว็บแอปพลิเคชันที่มั่นคงปลอดภัย รวมถึงศักยภาพในการแก้ไขช่องโหว่ของโค้ดที่ AI สร้างขึ้นเอง

ดังนั้น งานวิจัยนี้จึง มุ่งเน้นไปที่การประเมินความสามารถของ Generative AI ในการพัฒนาเว็บแอปพลิเคชันที่มีความมั่นคงปลอดภัย รวมถึง ความสามารถในการปรับปรุงและแก้ไขช่องโหว่ที่ตรวจพบ ซึ่งเป็นประเด็นที่ยังไม่ได้รับการศึกษาอย่างครอบคลุมมาก่อน นอกจากนี้ งานวิจัยนี้ยังมีการนำ Generative AI จำนวน 3 เครื่องมือ ได้แก่ ChatGPT, Gemini และ Claude มาใช้ในการทดลอง เพื่อ เปรียบเทียบความสามารถในการสร้างเว็บแอปพลิเคชันที่ปลอดภัยและการแก้ไขช่องโหว่ ซึ่งแตกต่างจากงานวิจัยก่อนหน้านี้ที่มัก เน้นศึกษาเฉพาะ ChatGPT เพียงตัวเดียว โดยที่การศึกษานี้จะช่วยให้เห็นถึงจุดแข็ง จุดอ่อน และประสิทธิภาพในการจัดการช่องโหว่ของ AI แต่ละตัว ซึ่งจะเป็นข้อมูลสำคัญในการประเมินศักยภาพของ Generative AI ในการพัฒนา เว็บแอปพลิเคชันที่มีความมั่นคงปลอดภัยในระดับที่ยอมรับได้ และสามารถนำไปใช้เป็นแนวทางสำหรับนักพัฒนา ในการเลือกใช้ Generative AI อย่างมีประสิทธิภาพและปลอดภัยมากขึ้น

บทที่ 3
วิธีดำเนินการวิจัย

การค้นคว้าอิสระเรื่อง การทดสอบประสิทธิภาพของ Generative AI ในการสร้างและแก้ไขช่องโหว่ในเว็บแอปพลิเคชันที่มีความมั่นคงปลอดภัย โดยใช้คำสั่งในภาษาไทย จึงมีการกำหนดวางแผนโครงสร้างการทำงานของเว็บแอปพลิเคชันที่จะให้ Generative AI สร้างขึ้น และขั้นตอนวิธีดำเนินการวิจัยดังนี้

3.1 การกำหนดโครงสร้างของเว็บแอปพลิเคชัน

ทางผู้วิจัยจะกำหนดให้ Generative AI ทำการสร้างเว็บแอปพลิเคชัน 3 รูปแบบดังนี้

3.1.1 ระบบเว็บแอปพลิเคชันหน้าเว็บเพจ

ให้ Generative AI สร้างเว็บเพจมา 1 หน้าโดยจะมีการเพิ่มให้มีการตกแต่งจัดรูปแบบความสวยงามเพิ่มขึ้น โดยใช้ HTML หรือ CSS

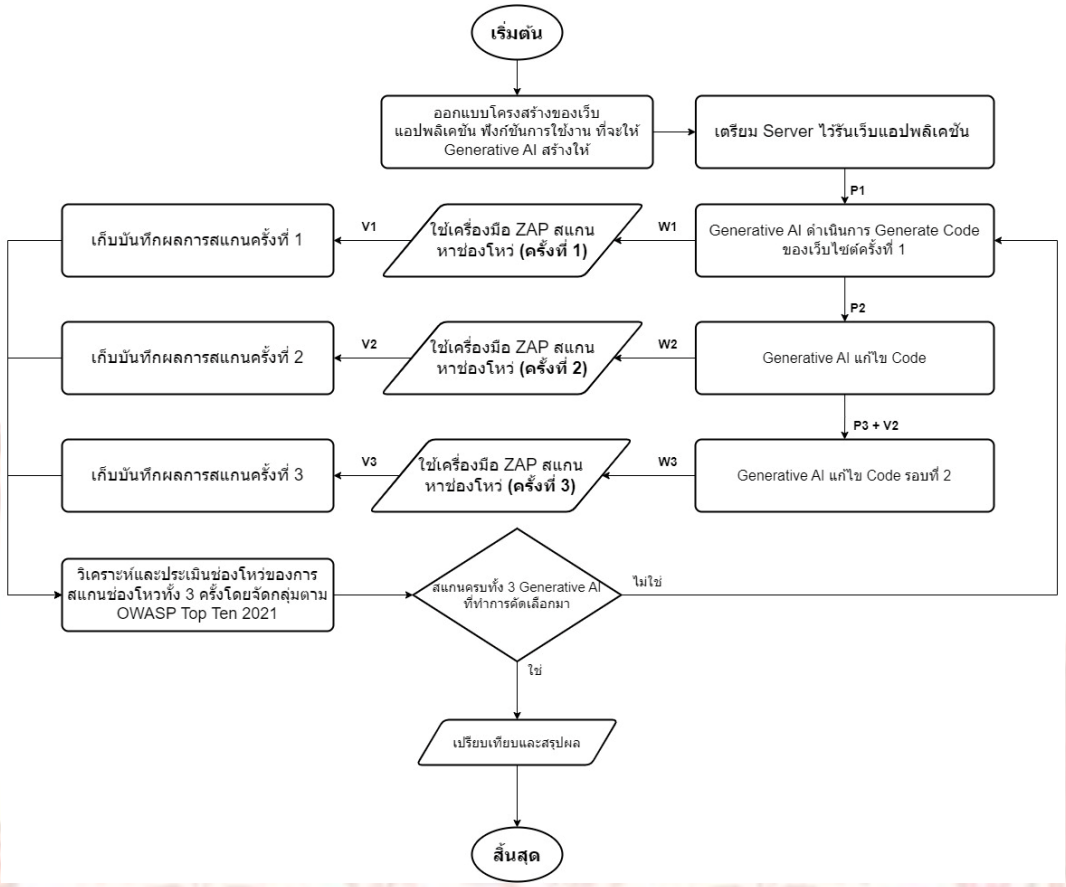
3.1.2 ระบบตรวจสอบตัวตน

ให้ Generative AI สร้างหน้าระบบตรวจสอบตัวตนผู้ใช้งาน (User Authentication System) โดยใช้ภาษา PHP, HTML และมีการเชื่อมต่อกับระบบฐานข้อมูล (MySQL Database) เพื่อตรวจสอบรหัสผู้ใช้งานและรหัสผ่านก่อนเข้าใช้งาน

3.1.3 ระบบจัดการไฟล์

ให้ Generative AI สร้างระบบอัพโหลดไฟล์รูปภาพ โดยมีช่องใส่ข้อมูลต่างๆของรูปภาพ, และจะมีอีกหน้าที่สามารถดึงข้อมูลรูปภาพและข้อความต่างๆขึ้นมา เพื่อที่จะสามารถแก้ไขข้อความหรือลบรูปภาพนั้นออกได้ (Function Insert/Update/Delete) ด้วยภาษา PHP และ HTML

3.2 ภาพรวมกระบวนการดำเนินงาน



ภาพที่ 3-1 ภาพรวมกระบวนการดำเนินงาน

<u>คำอธิบายตัวแปรในแผนภาพที่ 3-1</u>

P1 คือ Prompt ที่ส่งให้ Generative AI สร้างเว็บแอปพลิเคชัน

P2 คือ Prompt ที่ส่งให้ Generative AI แก้ไขช่องโหว่โดยไม่ระบุเจาะจง

P3 คือ Prompt ที่ส่งให้ Generative AI แก้ไขช่องโหว่โดยระบุเจาะจง

W1 คือ เว็บแอปพลิเคชันที่ Generative AI สร้างขึ้นมาให้ตอนเริ่มแรก

W2 คือ เว็บแอปพลิเคชันที่ Generative AI สร้างและมีการเพิ่มการแก้ไขช่องโหว่โดยไม่ระบุว่าเป็นช่องโหว่ใด ๆ

W3 คือ เว็บแอปพลิเคชันที่ Generative AI สร้างและมีการเพิ่มการแก้ไขช่องโหว่โดยระบุช่องโหว่ที่พบเจอไปด้วย

V1 คือ ผลลัพธ์ช่องโหวของการสแกน Code เริ่มต้นที่ได้จาก Generative AI

V2 คือ ผลลัพธ์ช่องโหวของการสแกน Code ที่ได้รับการปรับปรุงโดยไม่ระบุเจาะจงช่องโหว่

V3 คือ ผลลัพธ์ช่องโหวของการสแกน Code ที่ได้รับการปรับปรุงโดยระบุเจาะจงช่องโหว่ที่พบ

3.3 ขั้นตอนการดำเนินงานวิจัย

3.3.1 ทำการออกแบบโครงสร้างเว็บแอปพลิเคชัน ที่จะให้ Generative AI สร้างขึ้นมา

3.3.2 ทำการจัดเตรียมระบบเครื่องแม่ข่าย (Server) ไว้สำหรับรันใช้งานชุดคำสั่งหรือโค้ด (Code) ที่ได้มาจาก Generative AI

3.3.3 ส่งคำสั่ง (Prompt) ไปให้ยัง Generative AI เพื่อให้ทำการสร้างชุดคำสั่งหรือข้อความที่เขียนขึ้นโดยใช้ภาษาโปรแกรม (Programming Language) เพื่อสั่งให้คอมพิวเตอร์หรือระบบทำงานตามที่ผู้วิจัยต้องการ ตามที่ได้ออกแบบไว้ในข้อที่ 3.1.1

3.3.4 นำโค้ดที่ได้มาจาก Generative AI ไปทำการรันบนเครื่องแม่ข่ายที่เตรียมไว้ก่อนหน้า และผู้วิจัยจะทำการทดสอบเข้าใช้งานด้วยตนเองเบื้องต้นก่อน ว่าสามารถเข้าใช้งานได้ปกติตามได้ที่ออกแบบไว้

3.3.5 ทดสอบหาช่องโหว่ครั้งที่ 1 โดยใช้เครื่องมือ OWASP ZAP สแกน

3.3.6 เก็บบันทึกผลลัพธ์การสแกนหาช่องโหว่ของครั้งที่ 1

3.3.7 ส่งคำสั่งไปให้ยัง Generative AI เพื่อให้ทำการแก้ไขและสร้างโค้ดใหม่ โดยผู้วิจัยจะทำการส่งคำสั่งบอกตัว Generative AI ว่า ช่วยปรับปรุงแก้ไขให้โค้ดที่ได้มาเมื่อแรกนั้นให้มีความมั่นคงปลอดภัยมากขึ้น และเมื่อได้สร้างชุดคำสั่งใหม่แล้ว ผู้วิจัยจะทำการทดสอบเข้าใช้งานด้วยตนเองอีกครั้ง

3.3.8 ทดสอบหาช่องโหว่ครั้งที่ 2 โดยใช้เครื่องมือ OWASP ZAP สแกน

3.3.9 เก็บบันทึกผลลัพธ์การสแกนหาช่องโหว่ครั้งที่ 2

3.3.10 หลังจากได้ผลลัพธ์การสแกนหาช่องโหว่ครั้งที่ 2 ผู้วิจัยจะทำการส่งคำสั่งไปให้ยัง Generative AI อีกครั้งเพื่อให้ทำการแก้ไขและสร้างโค้ดใหม่ โดยครั้งนี้จะมีการระบุช่องโหว่ที่พบ

เจอจากการที่ทำการสแกนหาช่องโหว่แก่ Generative AI ไปด้วย และเมื่อได้สร้างชุดคำสั่งใหม่แล้ว ผู้วิจัยจะทำการทดสอบเข้าใช้งานด้วยตนเองอีกครั้ง

3.3.11 ทดสอบหาช่องโหว่ครั้งที่ 3 โดยใช้เครื่องมือ OWASP ZAP สแกน

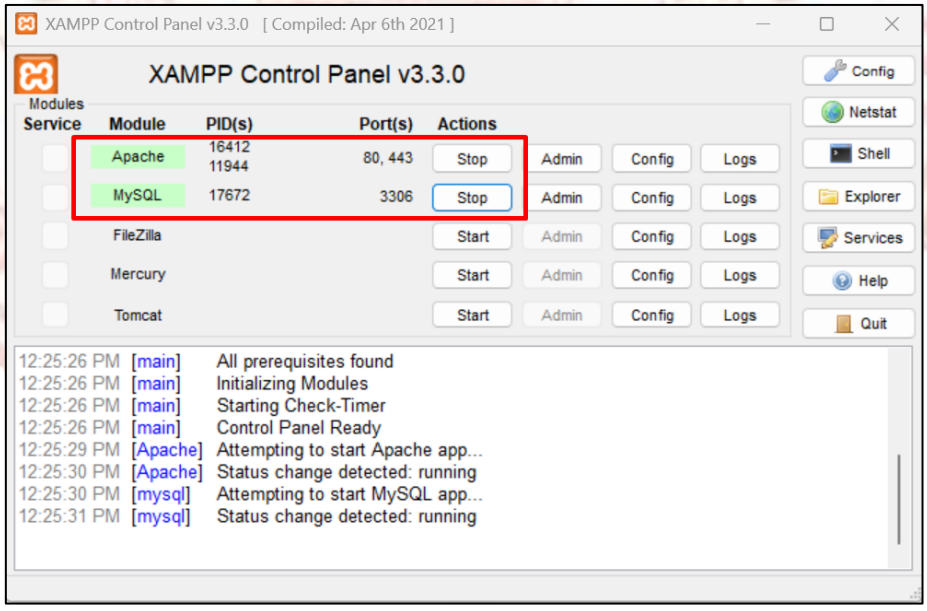
3.3.12 เก็บบันทึกผลลัพธ์การสแกนหาช่องโหว่ครั้งที่ 3

3.3.13 นำผลลัพธ์ของการสแกนหาช่องโหว่ทั้ง 3 มาวิเคราะห์และประเมินช่องโหว่โดยจัดกลุ่มตาม OWASP Top Ten 2021

3.3.14 สรุปผลการเปรียบเทียบผลลัพธ์ความแตกต่างของการสแกหาช่องโหว่ของ Generative AI ทั้งหมดที่เลือกมา

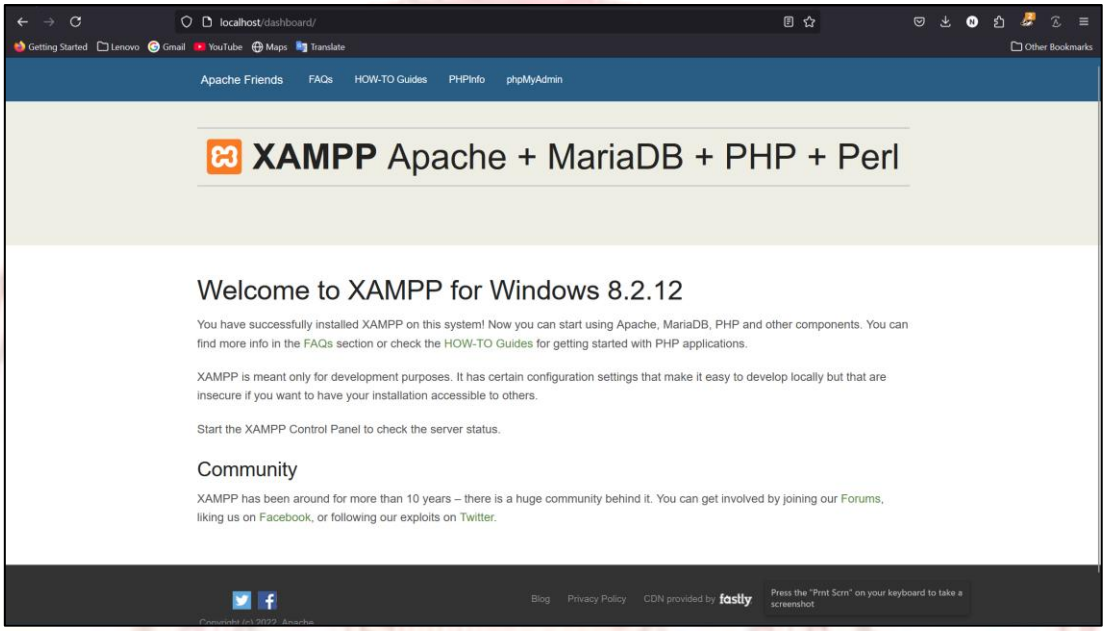
3.4 การเตรียมเครื่องมือและสภาพแวดล้อมการทดสอบ

3.4.1 ดำเนินการจำลองเว็บเซิร์ฟเวอร์ (Web Server) และระบบฐานข้อมูล (Database) โดย ติดตั้งโปรแกรม XAMPP [24] ซึ่งเป็นซอฟต์แวร์ที่รวมชุดโปรแกรมเพื่อใช้ทดสอบและพัฒนา เว็บแอปพลิเคชันที่ใช้ PHP และ MySQL

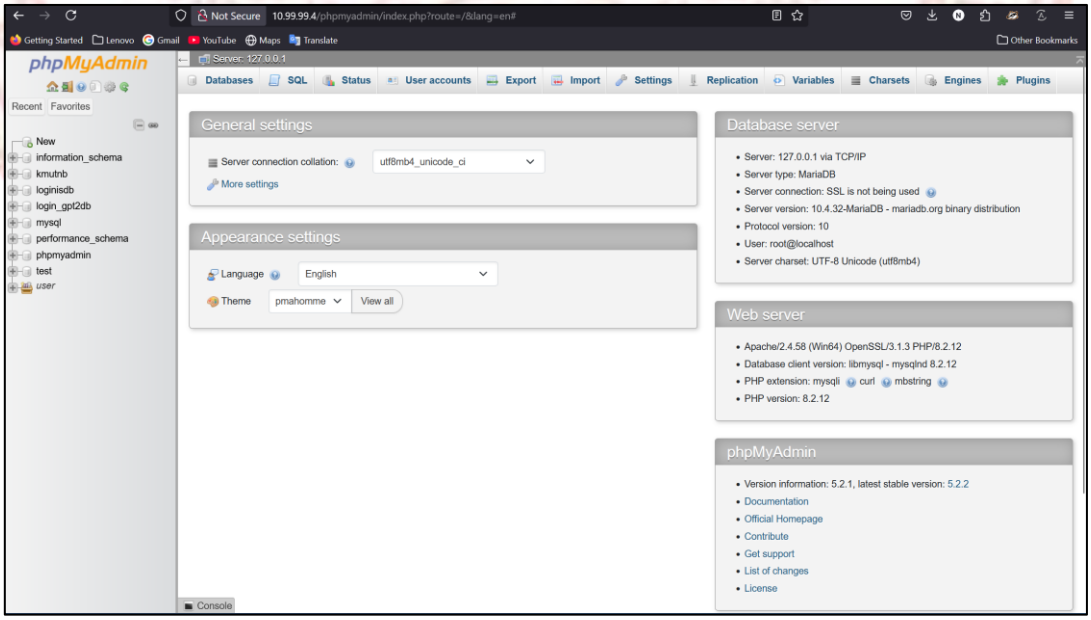


ภาพที่ 3-2 โปรแกรมจำลองเว็บเซิร์ฟเวอร์ XAMPP

จากนั้นติดตั้ง Apache [25] เว็บเซิร์ฟเวอร์ที่ใช้สำหรับรันโค้ด PHP และให้บริการหน้าเว็บ และติดตั้งระบบฐานข้อมูลที่ใช้เก็บข้อมูลของเว็บแอปพลิเคชัน ดังแสดงในภาพที่ 3-3 และ ภาพที่ 3-4



ภาพที่ 3-3 เว็บเซิร์ฟเวอร์ Apache

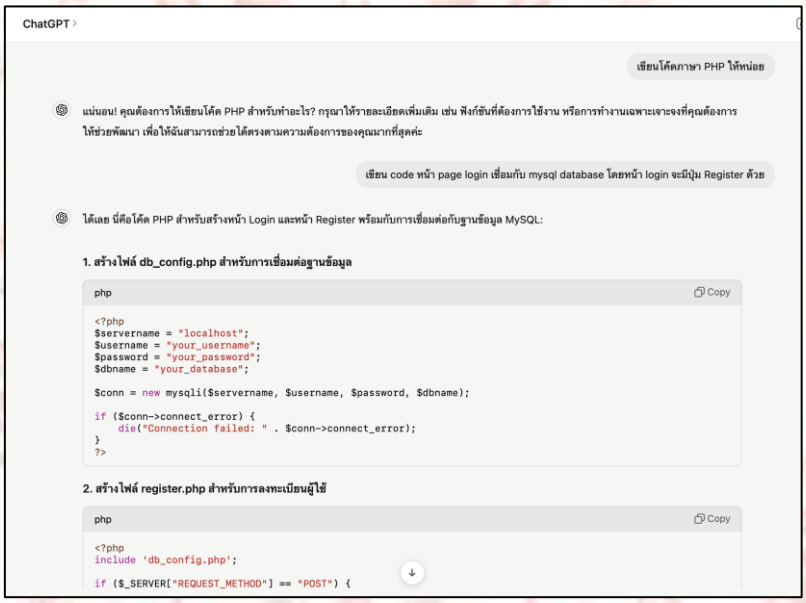


ภาพที่ 3-4 ระบบฐานข้อมูล MySQL

3.5 การสร้างเว็บแอปพลิเคชันด้วยเครื่องมือ AI

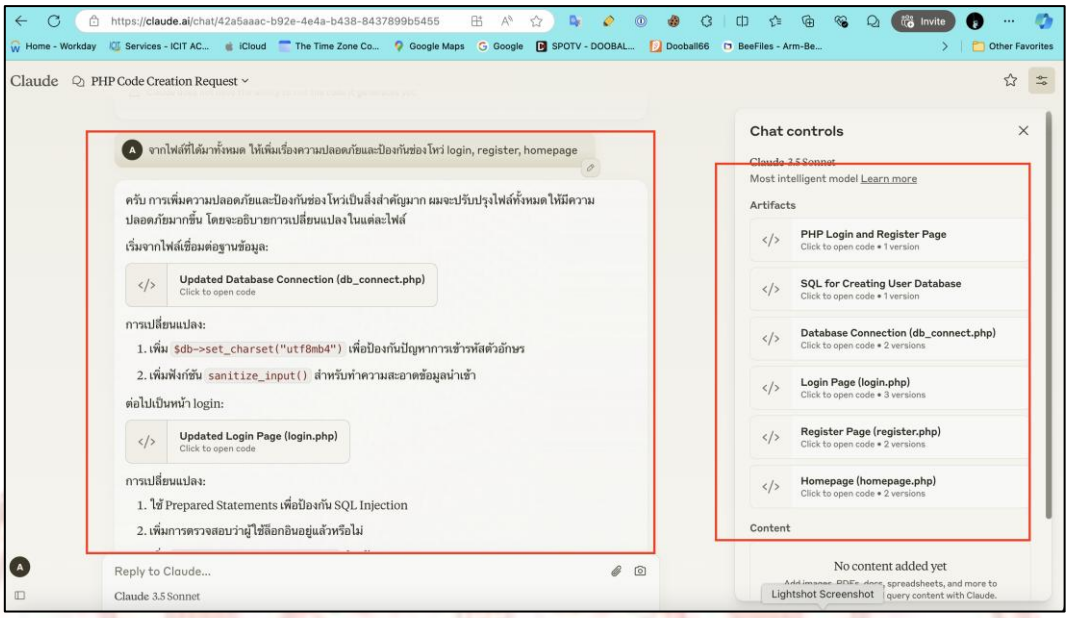
ผู้วิจัยจะดำเนินการส่งคำสั่งไปให้ยัง Generative AI เพื่อทำการสร้างเว็บแอปพลิเคชัน โดยจะระบุความต้องการต่าง ๆ ลงไปยัง prompt ให้แก่ AI เช่น ต้องการให้หน้าเว็บเพจนี้มีข้อมูลอะไรบ้าง หรือจะมีปุ่มที่เชื่อมโยงไปยังหน้าเพจอื่น ๆ หรือไม่ เพื่อให้ Generative AI สร้าง code PHP มาให้ แล้วนำไปทดสอบใช้งานบนเว็บเซิร์ฟเวอร์ที่ได้เตรียมไว้

อย่างไรก็ตาม ค่าพารามิเตอร์ต่าง ๆ เช่น ค่า temperature ซึ่งหมายถึงค่าที่ใช้ในการควบคุมสุ่มข้อความ (ถ้าค่าต่ำแม่นยำ คาดเดาได้ ถ้าค่าสูงหลากหลาย สร้างสรรค์ และคาดเดายาก) ที่ใช้ในงานวิจัยนี้เป็นค่า default ทั้งหมด



ภาพที่ 3-5 ตัวอย่างการส่ง prompt โต้ตอบกับเครื่องมือ AI (ChatGPT)

หลังจากที่ได้ทำการโต้ตอบ จนกระทั่งเครื่องมือ AI สามารถสร้าง code ให้ได้เสร็จตรงตามความต้องการแล้วหน้าจอของตัวเครื่องมือ AI จะทำการสรุป code ทั้งหมดที่ได้ทำการสร้างขึ้นมาให้แก่ผู้ใช้งาน ดังรูปภาพที่ 3-6 และภาพที่ 3-7 ซึ่งการแสดงผลของแต่ละเครื่องมือจะมีความต่างกันออกไป ขึ้นอยู่กับ UX (User Experience) และ UI (User Interface) ของแต่ละค่ายของ AI

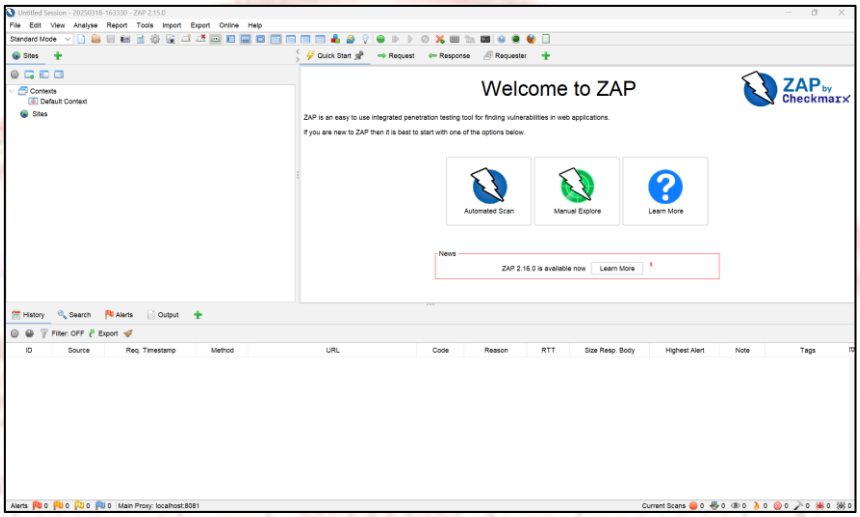


ภาพที่ 3-6 ตัวอย่างการแสดงผลลัพธ์ code ที่สร้างมาได้ของ Claude.ai

ภาพที่ 3-7 ตัวอย่างการแสดงผลลัพธ์ code ที่สร้างมาได้ของ Gemini

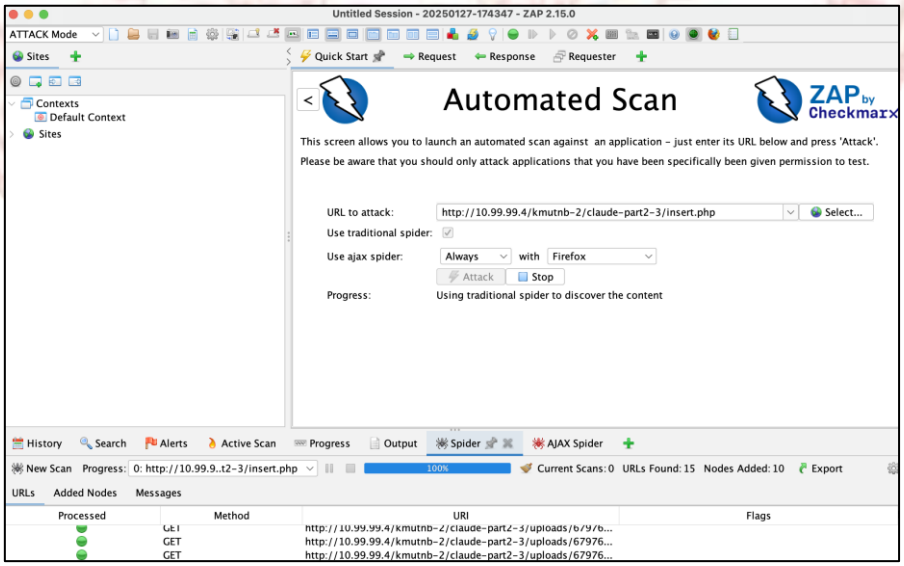
3.6 การทดสอบสแกนหาช่องโหว่

ขั้นตอนนี้ หลังจากผู้วิจัยได้นำ code ที่ได้จากเครื่องมือ Generative AI ไปรันทดสอบใช้งานบนเครื่องเว็บเซิร์ฟเวอร์ ผู้วิจัยจะนำโปรแกรม OWASP ZAP มาใช้สแกนช่องโหว่ของเว็บแอปพลิเคชันนั้น ๆ เพื่อทำการเก็บและวิเคราะห์ผลลัพธ์การวิจัย

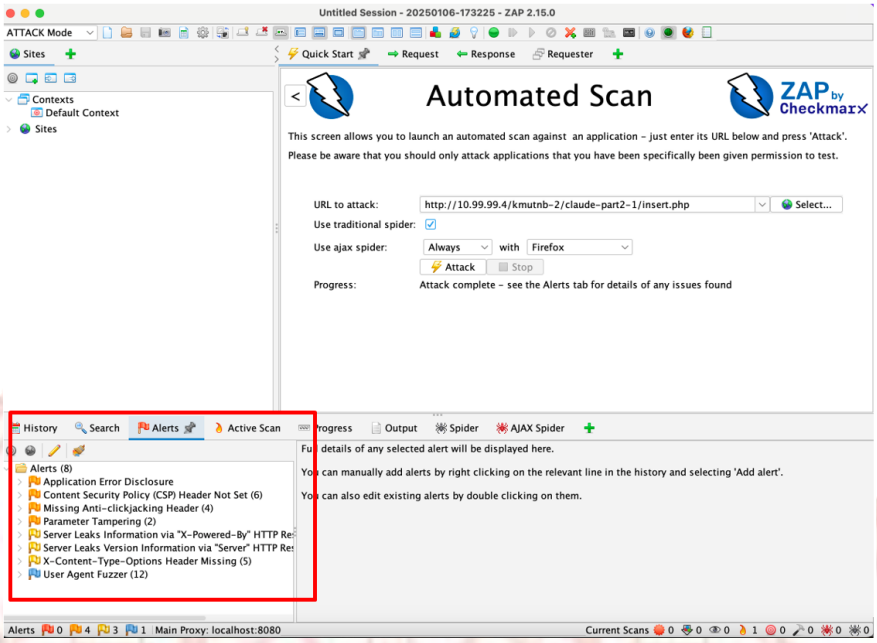


ภาพที่ 3-8 โปรแกรมสแกนช่องโหว่ OWASP ZAP

ผู้วิจัยจะทำการระบุข้อมูล Path ที่อยู่ของเว็บแอปพลิเคชัน ที่ต้องการจะสแกนหาช่องโหว่ไปให้แก่โปรแกรม OWASP ZAP จากนั้นกดปุ่ม Attack เพื่อเริ่มดำเนินการสแกนช่องโหว่ ดังภาพตัวอย่างที่ 3-9

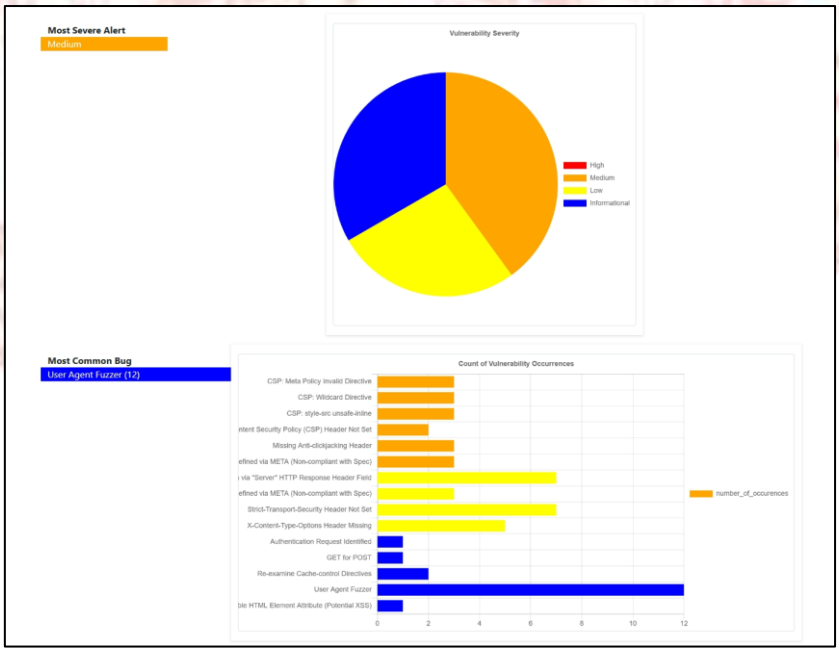


ภาพที่ 3-9 ตัวอย่างการสแกนหาช่องโหว่ด้วย OWASP ZAP



ภาพที่ 3-10 ผลลัพธ์การสแกนช่องโหว่โดย OWASP ZAP

เมื่อโปรแกรมสแกนเสร็จเรียบร้อยในแทป Alerts จะแสดงข้อมูลช่องโหว่ที่พบ สามารถกดเพื่อดูรายละเอียดเพิ่มเติมเกี่ยวกับแต่ละช่องโหว่ที่ตรวจพบได้ หลังจากนั้นผู้วิจัยจะทำการ Generate Report ออกมาจากตัวโปรแกรม ดังภาพที่ 3-11



ภาพที่ 3-11 ตัวอย่าง Scanning Report

3.7 การประเมินผลให้คะแนน

ขั้นตอนนี้ ผู้วิจัยจะใช้วิธีการ Numerical Rating Scale (NRS) เพื่อใช้ในการแปลงค่าวัดและให้คะแนนตามระดับ (Scale) ที่กำหนด เพื่อให้มีความสอดคล้องกับ ระดับช่องโหว่ที่พบ โดยผู้วิจัยสามารถกำหนดค่าได้ดังตารางที่ 3-1

ตารางที่ 3-1 ผลการแปลงค่าด้วยวิธีการ Numerical Rating Scale (NRS)

คะแนน	ความหมาย
3	ช่องโหว่ระดับรุนแรงสูง (High severity)
2	ช่องโหว่ระดับรุนแรงปานกลาง (Medium severity)
1	ช่องโหว่ระดับรุนแรงต่ำ (Low severity)

จากนั้นผู้วิจัยดำเนินการใช้สูตร Weighted Sum Model [26] คำนวณค่าผลรวมคะแนนระดับความรุนแรงของช่องโหว่ทั้ง 3 ระบบ โดยมีสูตรคำนวณดังสมการต่อไปนี้

$$Score_{system\ n} = (3\times High_{system\ n})+(2\times Medium_{system\ n})+(1\times Low_{system\ n}) \quad (3\text{-}1)$$

โดยที่

High คือ ผลรวมของจำนวนช่องโหว่ระดับรุนแรงสูง

Medium คือ ผลรวมของจำนวนช่องโหว่ระดับรุนแรงสูง

Low คือ ผลรวมของจำนวนช่องโหว่ระดับรุนแรงสูง

$$Total\ Score_{ChatGPT} = Score_{ChatGPT\ S1} + Score_{ChatGPT\ S2} + Score_{ChatGPT\ S3} \quad (3\text{-}2)$$

$$Total\ Score_{Gemini} = Score_{Gemini\ S1} + Score_{Gemini\ S2} + Score_{Gemini\ S3} \quad (3\text{-}3)$$

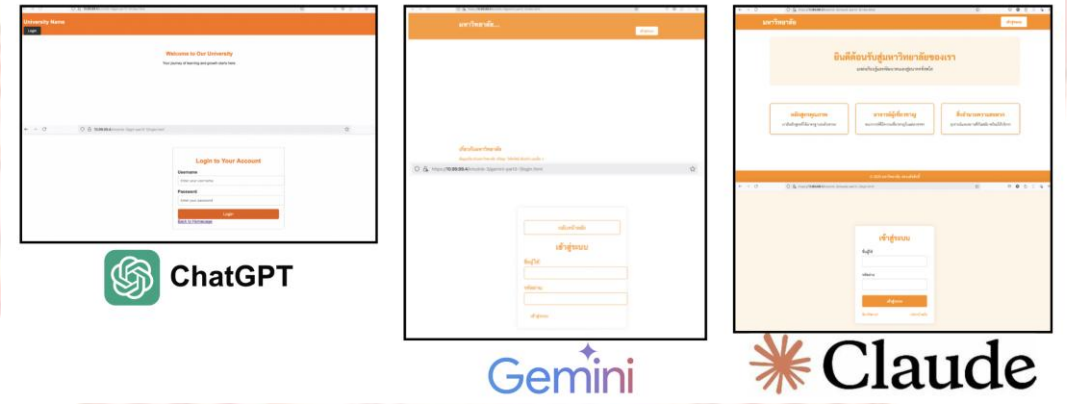
$$Total\ Score_{Claude} = Score_{Claude\ S1} + Score_{Claude\ S2} + Score_{Claude\ S3} \quad (3\text{-}4)$$

วิธีการเปรียบเทียบคะแนนของ AI แต่ละตัว เครื่องมือตัวไหนที่มีค่าคะแนนน้อยที่สุด ถือว่ามี ความสามารถในการแก้ไขช่องโหว่ได้มากที่สุด

บทที่ 4
ผลการวิจัย

จากการวิจัยเรื่องการทดสอบประสิทธิภาพของ Generative AI ในการสร้างและแก้ไขช่องโหว่บนเว็บแอปพลิเคชัน โดยจะทำการศึกษาความสามารถของเครื่องมือ Generative AI 3 ตัว คือ ChatGPT, Gemini และ Claude ในการสร้างและแก้ไขช่องโหว่ต่าง ๆ ที่พบในชุดคำสั่งที่ได้สร้างขึ้นมา

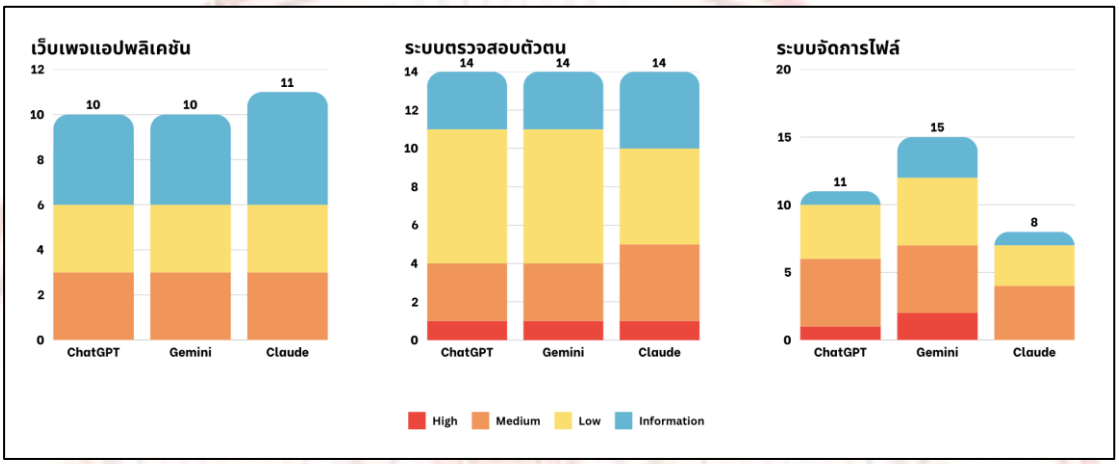
ซึ่งทั้งสามเครื่องมือที่นำมาใช้สามารถสร้างและจัดรูปแบบเว็บเพจให้เป็นไปตามคำสั่งที่กำหนดได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม แต่ละเครื่องมือมีแนวทางเฉพาะในการพัฒนาโค้ด ซึ่งแตกต่างกันในหลายแง่มุม โดยเฉพาะด้านโครงสร้าง HTML และการจัดการ CSS ส่งผลให้เว็บเพจที่ได้จากแต่ละเครื่องมือมีลักษณะเฉพาะตัว ทั้งรูปแบบการนำเสนอและวิธีการจัดการองค์ประกอบต่าง ๆ ภายในหน้าเว็บ ดังแสดงในภาพที่ 4-1



ภาพที่ 4-1 ผลลัพธ์ความแตกต่างของหน้าเว็บเพจที่สร้างโดยเครื่องมือ AI ทั้งสาม

4.1 ช่องโหว่ที่พบหลังจากสแกนครั้งที่ 1

จากการตรวจสอบ เว็บแอปพลิเคชันทั้ง 3 รายการ ที่พัฒนาโดย Generative AI ผ่านการสแกนช่องโหว่ด้วย OWASP ZAP พบว่ามี ช่องโหว่ที่ตรวจพบหลังทำการสแกนครั้งที่ 1 ดังแสดงในรูปภาพที่ 4-2



ภาพที่ 4-2 ภาพผลลัพธ์การสแกนครั้งที่ 1

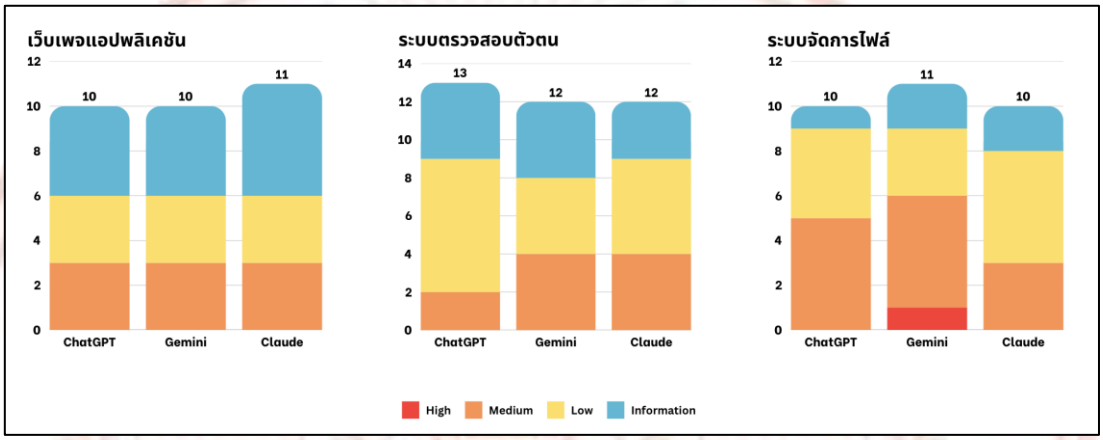
การสแกนช่องโหว่ครั้งแรกพบปัญหาด้านความมั่นคงปลอดภัยในหลายระดับ ตั้งแต่ช่องโหว่ร้ายแรง เช่น SQL Injection [27] และ Cross-Site Scripting (XSS) Persistent [28] ซึ่งอาจเปิดโอกาสให้ผู้โจมตีเข้าถึงหรือเปลี่ยนแปลงข้อมูลได้ โดยสองช่องโหว่ร้ายแรงนี้จะพบในระบบตรวจสอบตัวตนและระบบจัดการไฟล์ ไปจนถึงช่องโหว่ที่เกี่ยวข้องกับการตั้งค่าระบบ เช่น การขาด Anti-CSRF Token, Content Security Policy (CSP), X-Frame-Options และ การเปิดเผยข้อมูลเวอร์ชันเซิร์ฟเวอร์ ที่อาจเพิ่มความเสี่ยงต่อการโจมตี ซึ่งพบได้ในเกือบทุกระบบที่ให้ AI สร้างขึ้นมา และสิ่งที่น่าสนใจคือในแอปพลิเคชันระบบจัดการไฟล์ของ Claude ที่สร้างมานั้นไม่พบช่องโหว่ที่อยู่ในระดับรุนแรง (High Severity) เลย ซึ่งแตกต่างกับเครื่องมืออย่าง ChatGTP และ Gemini ที่พบเจอ

ส่วนในแอปพลิเคชันระบบเว็บเพจถึงแม้จะไม่พบช่องโหว่ระดับร้ายแรง แต่ก็ยังคงพบช่องโหว่ในระดับกลางจำพวก เช่น Content Security Policy (CSP), Server Leaks Version Information ที่ยังคงเป็นส่วนที่อันตรายต่อระบบด้วยเช่นกัน

4.2 ช่องโหว่ที่พบหลังจากสแกนครั้งที่ 2

ก่อนการสแกนรอบที่ 2 ผู้วิจัยได้สร้าง prompt เพื่อให้ Generative AI แต่ละเครื่องมือรับทราบว่าโค้ดที่สร้างขึ้นมีช่องโหว่ และขอ ให้ดำเนินการแก้ไข โดยไม่ได้ระบุรายละเอียดของช่องโหว่ที่ตรวจพบ ในขั้นตอนนี้เริ่มต้นทางผู้วิจัยได้ส่งคำสั่งที่ใช้ในขั้นตอนนี้คือ

“ให้ช่วยเพิ่ม code ในเรื่องความปลอดภัยเพื่อป้องกันช่องโหว่ต่าง ๆ เพิ่มให้หน่อย”

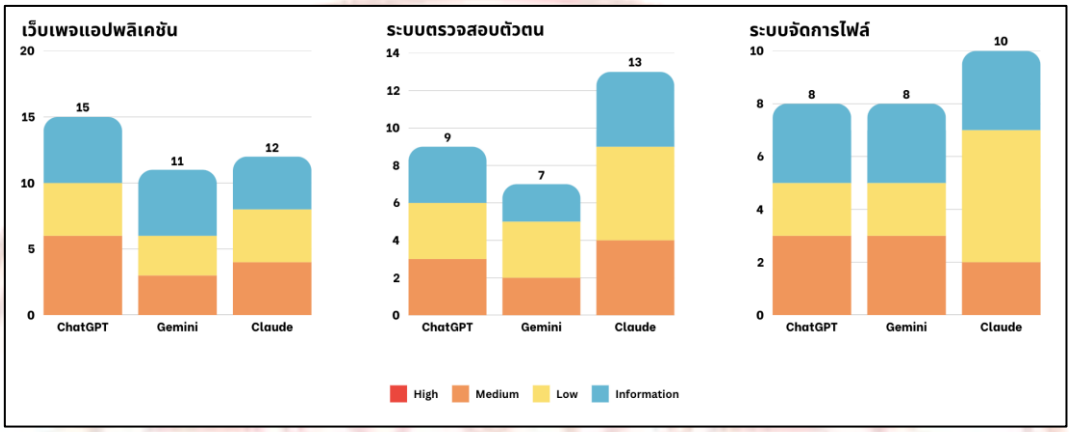


ภาพที่ 4-3 ภาพผลลัพธ์การสแกนครั้งที่ 2

ภายหลังการสแกนรอบที่ 2 พบว่าในแอปพลิเคชันเว็บเพจพบว่าไม่มีการเปลี่ยนแปลงเกิดขึ้น ช่องโหว่ที่พบเจอยังคงเท่าเดิม และยังคงไม่ได้รับการแก้ไขใด ๆ ส่วนในแอปพลิเคชันระบบตรวจสอบตัวตนและระบบจัดการไฟล์ ช่องโหว่ระดับ High Severity เช่น SQL Injection และ Persistent Cross-Site Scripting (XSS) ได้รับการแก้ไขบางส่วนในระบบของ ChatGPT และ Claude แต่ยังคงพบปัญหาดังกล่าวใน Gemini นอกจากนี้ ยังพบว่ามีช่องโหว่ใหม่เกิดขึ้น เช่น CSP: Wildcard Directive ในแอปพลิเคชันของ ChatGPT และบางช่องโหว่จากการสแกนรอบแรกยังคงหลงเหลืออยู่ สำหรับช่องโหว่ที่มีความร้ายแรงสูงอย่าง SQL Injection นั้น พบว่าทุกเครื่องมือสามารถแก้ไขได้เกือบทั้งหมด ยกเว้นระบบจัดการไฟล์ของ Gemini ซึ่งยังคงมีช่องโหว่ร้ายแรงดังกล่าวเหลืออยู่ โดยช่องโหว่ที่เกี่ยวกับจำพวก Content Security Policy (CSP) Header, Server Leaks Information ทั้งสามเครื่องมือยังคงพบปัญหานี้อยู่ ส่วนปัญหาช่องโหว่ที่เกี่ยวข้องกับ Cookie No HttpOnly Flag และ Cookie without SameSite Attribute เครื่องมือ Gemini สามารถจัดการแก้ไขช่องโหว่นี้ได้ดีที่สุด ดังที่แสดงในรูปภาพที่ 4-3

4.3 ช่องโหว่ที่พบหลังจากสแกนครั้งที่ 3

การสแกนช่องโหว่ครั้งที่ 3 นี้ผู้วิจัยได้ทำการส่ง prompt ให้แก่เครื่องมือ Generative AI โดยจะแจ้งรายละเอียดของช่องโหว่ต่าง ๆที่พบเจอจากการสแกนของรอบที่ 2 เพื่อให้แต่ละเครื่องมือ ได้ทำการแก้ไข Code อีกครั้งหนึ่ง โดยมีผลลัพธ์ดังรูปภาพที่ 4-4



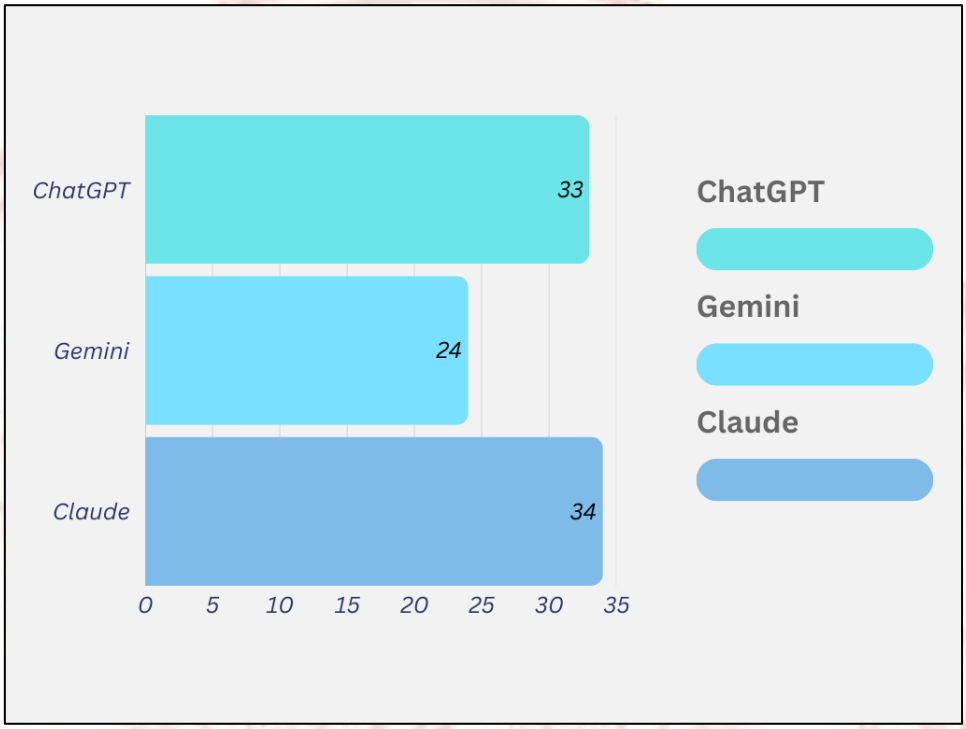
ภาพที่ 4-4 ภาพผลลัพธ์การสแกนครั้งที่ 3

ในระบบเว็บแอปพลิเคชันเว็บเพจพบว่า AI ทั้งสามสามารถแก้ไขช่องโหว่ได้แค่ช่องโหว่เดียวคือ Absence of Anti-CSRF Tokens อีกทั้งยังมีช่องโหว่ระดับปานกลาง (Medium) ที่เกิดขึ้นมาใหม่ด้วย โดย ChatGPT มีช่องโหว่ระดับนี้เกิดใหม่ขึ้นมาสูงสุดถึง 4 ช่องโหว่ และข้อสังเกตช่องโหว่ใหม่พี่พบจะเป็นช่องโหว่ประเภท Content Security Policy (CSP) หลายชนิด

ในส่วนของระบบตรวจสอบตัวตนและระบบจัดการไฟล์ผลลัพธ์ของเครื่องมือ AI สามารถกำจัดช่องโหว่ระดับ High Severity ได้ทั้งหมด แต่ยังคงเหลือช่องโหว่ระดับ Medium อยู่หลายรายการที่ไม่สามารถกำจัดไปได้อาทิเช่น ช่องโหว่จำพวก Content Security Policy (CSP) และ Cookie Security และช่องโหว่ที่อยู่ในระดับต่ำ (Low) ก็ถูกแก้ไขจนจำนวนเหลือน้อยลง จะมีคงเหลือที่ไม่สามารถแก้ไขได้คือพวกช่องโหว่การรั่วไหลข้อมูลต่าง ๆของ Server (Server Leaks Information) และ X-Content-Type-Option Header Missing ซึ่ง ทั้ง 3 เครื่องมือสามารถแก้ไขช่องโหว่ต่าง ๆได้ใกล้เคียงกัน ซึ่งโดยรวมทั้งสามเครื่องมือสามารถลดจำนวนช่องโหว่ที่มีความอันตรายลงได้ ดังแสดงในรูปภาพที่ 4-4

4.4 ผลคะแนนประเมินช่องโหว่

ผู้วิจัยนำผลลัพธ์ของการสแกนช่องโหว่ครั้งที่ 3 ไปดำเนินการคำนวณเพื่อหาค่าคะแนน และนำมาเปรียบเทียบ ประสิทธิภาพ ของทั้งสามเครื่องมือที่นำมาศึกษา



ภาพที่ 4-5 ผลคะแนนหลังให้เครื่องมือ AI แก้ไขช่องโหว่

ภาพที่ 4-5 แสดงผลรวมคะแนนของช่องโหว่ที่ยังคงเหลืออยู่ในแต่ละเครื่องมือ พบว่า Claude เป็นเครื่องมือที่ยังคงมีช่องโหว่หลงเหลือมากที่สุด โดยมีคะแนนอยู่ที่ 34 คะแนน รองลงมาคือ ChatGPT ซึ่งมีคะแนนอยู่ที่ 33 คะแนน แสดงให้เห็นว่า Claude และ ChatGPT มีความสามารถในการแก้ไขช่องโหว่ในระดับที่ใกล้เคียงกัน ในขณะที่ Gemini มีคะแนนต่ำที่สุด โดยมีคะแนนอยู่ที่ 24 คะแนน

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

จากผลการค้นคว้าอิสระเรื่อง การทดสอบประสิทธิภาพของ Generative AI ในการสร้างและแก้ไขช่องโหว่บนเว็บแอปพลิเคชัน สามารถสรุป อภิปราย และได้ข้อเสนอแนะดังต่อไปนี้

5.1 สรุปผลการวิจัย

งานวิจัยนี้มีเป้าหมายเพื่อประเมินประสิทธิภาพของ Generative AI ในการสร้างและปรับปรุงความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน โดยทดสอบผ่านการใช้ ChatGPT, Gemini และ Claude ในการพัฒนาเว็บเพจที่มีฟังก์ชันการทำงานสำคัญ เช่น ระบบตรวจสอบตัวตน (Login Page) และระบบจัดการไฟล์ที่มีการเข้าไปเกี่ยวข้องระบบฐานข้อมูล ซึ่งพัฒนาด้วย PHP, HTML และ MySQL Database จากนั้นทำการสแกนหาช่องโหว่ของระบบที่พัฒนาโดย AI ด้วย OWASP ZAP เพื่อประเมินจุดอ่อนของโค้ดที่ AI สร้างขึ้น และความสามารถของ AI ในการปรับปรุงช่องโหว่เมื่อได้รับคำสั่ง กระบวนการวิจัยนี้แบ่งออกเป็นสามรอบ ได้แก่ การสแกนรอบแรก ซึ่งเป็นการประเมินความมั่นคงปลอดภัยของโค้ดดั้งเดิมที่สร้างโดย AI โดยไม่มีการปรับปรุงเพิ่มเติม การสแกนรอบที่สอง ซึ่งให้ AI ปรับปรุงด้านความมั่นคงปลอดภัยโดยไม่มีการระบุช่องโหว่แบบเฉพาะเจาะจง และ การสแกนรอบที่สาม ซึ่งเป็นการให้ AI แก้ไขช่องโหว่ตามผลการสแกนรอบก่อนหน้า

ผลการสแกนรอบแรก พบว่าระบบที่สร้างขึ้นมีช่องโหว่สำคัญหลายรายการ โดยเฉพาะช่องโหว่ที่สามารถถูกใช้ในการโจมตีเว็บแอปพลิเคชัน เช่น SQL Injection (SQLi), Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Parameter Tampering เป็นต้น นอกจากนี้ ยังพบปัญหาด้าน Security Headers ซึ่งเป็นการขาดมาตรการป้องกันที่สำคัญ เช่น Missing Content Security Policy (CSP) และ Clickjacking, Missing X-Frame-Options อาจทำให้เว็บถูกโจมตีแบบ Clickjacking, Cookie Security Issues คุกกี้ไม่มีการตั้งค่า HttpOnly, Secure และ SameSite Attribute ทำให้สามารถถูกขโมยหรือใช้เพื่อโจมตีแบบ CSRF ได้

เมื่อเข้าสู่การสแกนรอบที่สอง ซึ่งเป็นการให้ AI ปรับปรุงด้านความมั่นคงปลอดภัยแบบทั่วไป พบว่า AI สามารถแก้ไขช่องโหว่บางรายการ เช่น SQL Injection ได้สำเร็จในบางเครื่องมือ อย่างไรก็ตาม การปรับปรุงแบบทั่วไปยังคงไม่สามารถกำจัดช่องโหว่ได้ทั้งหมด และ

ในบางกรณี การแก้ไขของ AI อาจสร้างช่องโหว่ใหม่ขึ้นมาแทน เช่น CSP ที่กำหนดค่าแบบ Wildcard ซึ่งไม่ปลอดภัย และการขาด SameSite Attribute ในคุกกี้ ซึ่งอาจเพิ่มความเสี่ยงต่อการโจมตีแบบ Cross-Site Scripting (XSS) และ Clickjacking

ในรอบที่สาม หลังจากแจ้งช่องโหว่ที่พบให้ AI แก้ไขแบบเฉพาะเจาะจง พบว่าคุณภาพของโค้ดดีขึ้นอย่างมีนัยสำคัญ โดยพบว่าช่องโหว่ CSRF ถูกกำจัดทั้งหมด และ CSP ได้รับการกำหนดค่าให้ปลอดภัยยิ่งขึ้น อย่างไรก็ตาม ยังมีปัญหาบางรายการที่ไม่ได้รับการแก้ไขอย่างสมบูรณ์ เช่น Security Headers และ Cookie Security ซึ่งเป็นปัจจัยสำคัญในการป้องกันการโจมตีทางไซเบอร์ และจากผลรวมคะแนนของช่องโหว่ที่ยังคงเหลืออยู่ในแต่ละเครื่องมือพบว่า Gemini เป็นเครื่องมือที่สามารถลดจำนวนช่องโหว่ได้มากที่สุด เมื่อเปรียบเทียบกับ ChatGPT และ Claude ที่ผลคะแนนของช่องโหว่ที่สูงกว่า

โดยสรุปนอกจากความสามารถในการสร้างเว็บเพจแล้ว AI ยังสามารถช่วยปรับปรุงโค้ดให้มีความมั่นคงปลอดภัยมากขึ้นได้ด้วย โดยสามารถวิเคราะห์และแก้ไขจุดอ่อนที่อาจนำไปสู่ช่องโหว่ด้านความมั่นคงปลอดภัย นอกจากนี้ AI ยังสามารถตอบสนองต่อคำสั่งของผู้วิจัยได้อย่างยืดหยุ่น แม้ว่าผู้วิจัยจะไม่ได้ใช้ศัพท์ทางเทคนิคอย่างละเอียดในทุกครั้ง เครื่องมือ AI ก็ยังสามารถเข้าใจและประมวลผลตามเจตนารมณ์ของคำสั่งที่ส่งไปได้ และ Generative AI สามารถช่วยพัฒนาเว็บแอปพลิเคชันที่ปลอดภัยได้ในระดับหนึ่ง แต่ยังคงมีข้อจำกัดในการจัดการด้านความมั่นคงปลอดภัยอย่างสมบูรณ์ กล่าวคือ ไม่สามารถกำจัดช่องโหว่ที่พบทั้งหมดให้หายไปได้ ดังนั้นการใช้เครื่องมือ AI ควรทำควบคู่ไปกับ การสแกนช่องโหว่และการตรวจสอบโดยผู้เชี่ยวชาญ เพื่อให้มั่นใจว่าระบบมีความมั่นคงปลอดภัยเพียงพอสำหรับการใช้งานจริง

5.2 อภิปรายผลการวิจัย

ในการศึกษาวิจัยครั้งนี้ ผู้วิจัยได้ทำการทดสอบและเปรียบเทียบประสิทธิภาพของเครื่องมือ Generative AI ทั้งสามแพลตฟอร์ม ได้แก่ ChatGPT, Gemini และ Claude โดย ใช้ค่าพารามิเตอร์ Temperature ตามค่าเริ่มต้น (Default) ของแต่ละเครื่องมือ โดยไม่มีการปรับแต่งเพิ่มเติม เนื่องจากค่า Temperature เป็นค่าพารามิเตอร์สำคัญที่มีผลต่อลักษณะของคำตอบที่สร้างขึ้นโดย AI โดยทำหน้าที่ควบคุมระดับความหลากหลายและความคาดเดาไม่ได้ของผลลัพธ์ ค่าที่ต่ำ จะทำให้ AI เลือกคำที่มีโอกาสสูงสุดในการตอบ ส่งผลให้ได้คำตอบที่แม่นยำและสม่ำเสมอ ขณะที่ค่าที่สูงขึ้นจะเพิ่มความหลากหลายของคำตอบ ทำให้ได้ผลลัพธ์ที่สร้างสรรค์และแตกต่างกันมากขึ้น

อย่างไรก็ตาม ค่าพารามิเตอร์ Temperature เริ่มต้นของแต่ละเครื่องมืออาจแตกต่างกันขึ้นอยู่กับการออกแบบของแต่ละผู้พัฒนา เพื่อให้เกิดความเป็นกลางในการทดสอบ ผู้วิจัยจึงเลือกใช้ค่าพารามิเตอร์ Temperature ตามค่าดั้งเดิมของแต่ละแพลตฟอร์ม โดยไม่มีการปรับเปลี่ยนเพิ่มเติม เพื่อให้สามารถประเมินประสิทธิภาพของ Generative AI แต่ละตัวภายใต้เงื่อนไขการใช้งานปกติได้อย่างเป็นกลาง โดยผลการทดสอบความสามารถของแต่ละเครื่องมือสามารถอภิปรายผลได้ดังนี้

5.2.1 ประสิทธิภาพของ Generative AI ในการสร้างเว็บแอปพลิเคชัน

จากการทดลองพบว่า Generative AI สามารถสร้างโครงสร้างเว็บแอปพลิเคชันได้ตรงตามคำสั่ง (Prompt) ที่กำหนด รวมถึงการใช้ HTML, CSS และ PHP ได้อย่างถูกต้อง อย่างไรก็ตามพบว่ามี ความแตกต่างในเทคนิคการสร้างโค้ดของแต่ละ AI ซึ่งอาจมีผลต่อช่องโหว่ที่เกิดขึ้นในภายหลัง

5.2.2 ความสามารถในการแก้ไขช่องโหว่

ผลการทดลองแสดงให้เห็นว่า Generative AI สามารถแก้ไขช่องโหว่ได้ดียิ่งขึ้นเมื่อได้รับข้อมูลที่เจาะจง โดยเฉพาะในรอบที่สาม ซึ่งให้ข้อมูลช่องโหว่โดยตรง อย่างไรก็ตาม AI ยังไม่สามารถแก้ไขช่องโหว่ทั้งหมดได้อย่างสมบูรณ์ เช่น ปัญหาด้าน CSP, Secure Cookies และ Security Headers

5.2.3 ข้อจำกัดของ Generative AI

ถึงแม้ว่า Generative AI จะสามารถช่วยพัฒนาและแก้ไขโค้ดได้ แต่ก็ยังมีข้อจำกัด ได้แก่

5.2.3.1 Generative AI ไม่สามารถคาดการณ์ช่องโหว่ได้เองทั้งหมด หากไม่มีข้อมูลแนวทางการแก้ไขที่ชัดเจน ช่องโหว่บางส่วนอาจยังคงอยู่

5.2.3.2 ความแตกต่างระหว่างเครื่องมือที่อาจมีฐานความรู้ต่างกัน เช่น Gemini มีแนวโน้มที่จะแก้ไข CSRF Tokens ได้ดีกว่าเครื่องมืออื่น ขณะที่ ChatGPT ยังคงมีปัญหาด้าน Application Error Disclosure ซึ่งในบางสถานการณ์ เครื่องมือ AI ยังคงแก้ไขช่องโหว่ได้ไม่เท่ากัน ส่วน Claude นั้นมีความสามารถในการจัดการสร้างระบบที่ปลอดภัยและช่องโหว่ในระดับรุนแรงสูงได้ดีกว่าเครื่องมืออื่น ๆ เพราะมีในบางสถานการณ์ Claude ไม่พบช่องโหว่ในระดับสูงเลยถึงแม้จะเป็นเพียงการสแกนรอบแรกก็ตาม

5.2.3.3 การดำเนินการแก้ไขช่องโหว่บางอย่างอาจก่อให้เกิดช่องโหว่ใหม่ เช่น การตั้งค่า CSP ที่ผิดพลาดอาจทำให้เว็บแอปพลิเคชันเสี่ยงต่อ Cross-Site Scripting (XSS) ได้ ซึ่ง AI อาจจะไม่ได้มีการคิดคำนวณในเรื่องนี้มาก่อนว่าจะมีช่องโหว่ใหม่เกิดขึ้นมาหลังที่ได้ทำการอัพเดทตัว code จากคำสั่ง prompt ที่ผู้ใช้งานได้ส่งไป

5.3 ข้อเสนอแนะ

5.3.1 ข้อเสนอแนะสำหรับนักพัฒนา

แม้ว่า AI จะช่วยเร่งกระบวนการพัฒนา แต่ นักพัฒนายังคงต้องมีบทบาทสำคัญในการตรวจสอบและปรับปรุงโค้ด ข้อเสนอแนะต่อไปนี้เป็นแนวทางที่สามารถช่วยให้การใช้ Generative AI มีประสิทธิภาพมากขึ้น

5.3.1.1 ไม่ควรพึ่งพา AI เพียงอย่างเดียว AI สามารถสร้างโค้ดได้อย่างรวดเร็ว แต่ไม่สามารถรับประกันได้ว่าโค้ดนั้นจะปลอดภัย 100% นักพัฒนาควรตรวจสอบโค้ดด้วยตนเองเสมอ โดยเฉพาะในส่วนที่เกี่ยวข้องกับ การจัดการข้อมูลผู้ใช้, การยืนยันตัวตน (Authentication), และการเข้าถึงฐานข้อมูล นอกจากนี้ ควรนำเครื่องมือตรวจสอบช่องโหว่อัตโนมัติ เช่น OWASP ZAP, Burp Suite หรือ SonarQube มาช่วยเสริมการตรวจสอบ

5.3.1.2 ควรมีการสแกนช่องโหว่หลายรอบจากการทดลองพบว่า การให้ AI ปรับปรุงโค้ดโดยไม่ระบุช่องโหว่ที่ชัดเจน อาจยังคงมีช่องโหว่หลงเหลืออยู่ ดังนั้น นักพัฒนาควรดำเนินการ สแกนช่องโหว่หลายรอบ

5.3.1.3 ควรนำแนวทางความมั่งคงปลอดภัยของ OWASP มาปรับใช้ นักพัฒนาควรนำแนวทางเหล่านี้มาใช้เพื่อป้องกันช่องโหว่ที่อาจเกิดขึ้น เช่น การใช้ Anti-CSRF Tokens เพื่อป้องกัน Cross-Site Request Forgery (CSRF) หรือแม้แต่ การตั้งค่า Secure Cookies และ SameSite Attribute เพื่อป้องกันการขโมยเซสชันแนวทางเหล่านี้สามารถช่วยเสริมความมั่นคงปลอดภัยให้กับเว็บแอปพลิเคชันที่สร้างขึ้นโดย AI และลดโอกาสของการโจมตีจากแฮกเกอร์

5.3.2 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

แม้ว่าการศึกษานี้จะแสดงให้เห็นถึงศักยภาพและข้อจำกัดของ Generative AI ในการพัฒนาเว็บแอปพลิเคชัน แต่ยังมีหลายประเด็นที่ควรศึกษาเพิ่มเติมเพื่อให้สามารถใช้ AI ได้อย่างปลอดภัยและมีประสิทธิภาพมากขึ้น ดังนี้

5.3.2.1 ศึกษาความสามารถของ Generative AI รุ่นใหม่ๆ เนื่องจาก AI มีการพัฒนาอย่างต่อเนื่อง การศึกษาความสามารถของ Generative AI รุ่นใหม่จะช่วยให้เข้าใจว่าการอัปเดตเหล่านี้สามารถลดช่องโหว่ได้ดีขึ้นหรือไม่ นอกจากนี้ อาจมีการพัฒนา AI ที่ถูกออกแบบมาโดยเฉพาะสำหรับการสร้างโค้ดที่ปลอดภัย ซึ่งควรนำมาทดลองเปรียบเทียบกับ AI ทั่วไป

5.3.2.2 เปรียบเทียบผลลัพธ์ของ AI กับโค้ดที่มนุษย์เขียนการเปรียบเทียบคุณภาพของโค้ดที่สร้างขึ้นโดย AI กับโค้ดที่เขียนโดยนักพัฒนามืออาชีพ งานวิจัยในอนาคตอาจวิเคราะห์ว่า AI สามารถเขียนโค้ดได้ใกล้เคียงกับโค้ดของมนุษย์หรือไม่ โดยพิจารณาปัจจัยด้านความมั่นคงปลอดภัยของโค้ด (Security), ประสิทธิภาพของโค้ด (Performance), โครงสร้างโค้ดที่อ่านง่ายและแก้ไขได้ง่าย (Code Readability & Maintainability)

5.3.2.3 ศึกษาความเป็นไปได้ในการผสมผสาน AI กับเครื่องมือตรวจจับช่องโหว่อัตโนมัติ หากสามารถรวม Generative AI เข้ากับ เครื่องมือตรวจจับช่องโหว่อัตโนมัติ อาจช่วยเพิ่มประสิทธิภาพในการพัฒนาเว็บแอปพลิเคชันให้ปลอดภัยมากขึ้น ตัวอย่างเครื่องมือที่สามารถนำมาผสมผสาน เช่น Genertive AI ทำงานร่วมกันกับ OWASP ZAP สำหรับการตรวจจับช่องโหว่เว็บแอปพลิเคชัน หรือ Genertive AI ทำงานร่วมกันกับ Burp Suite สำหรับการวิเคราะห์ช่องโหว่และการทดสอบการเจาะระบบ (Penetration Testing) การวิจัยเพิ่มเติมในด้านนี้จะช่วยให้ AI สามารถตรวจสอบและแก้ไขช่องโหว่ได้อย่างแม่นยำยิ่งขึ้น

บรรณานุกรม

IEEE Spectrum. What is generative AI, and why is it suddenly everywhere? [ออนไลน์] 2568. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://spectrum.ieee.org/ai-mistakes-schneier

Vox. What is generative AI, and why is it suddenly everywhere? [ออนไลน์] 2566. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://www.vox.com/recode/2023/1/5/23539055/generative-ai-chatgpt-stable-diffusion-lensa-dall-e

Samia Kabir, David N. Udo-Imeh, Bonan Kou, and Tianyi Zhang. 2024. Is Stack Overflow Obsolete? An Empirical Study of the Characteristics of ChatGPT Answers to Stack Overflow Questions. In Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24). Association for Computing Machinery, New York, NY, USA, Article 935, 1–17.

Amazon. What is a Web Application? [ออนไลน์] 2567. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://aws.amazon.com/what-is/web-application/

OWASP. Top 10 Web Application Security Risks [ออนไลน์] 2564. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://owasp.org/www-project-top-ten/

ZAP. ZAP - Getting Started [ออนไลน์] 2020. [สืบค้นวันที่ 7 ธันวาคม 2566]. จาก https://www.zaproxy.org/getting-started/.

Burp Suite. Burp Suite's web vulnerability scanner [ออนไลน์] 2024. [สืบค้นวันที่ 7 ธันวาคม 2566]. จาก https://portswigger.net/burp/vulnerability-scanner

Nikto 2.5 [ออนไลน์] 2561. [สืบค้นวันที่ 7 ธันวาคม 2566]. จาก https://cirt.net/Nikto2

VISAI. ศักยภาพของ Generative AI ในการใช้งานทั่วไปและในภาคธุรกิจ [ออนไลน์] 2567. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://visai.ai/th/blogs/17/generative-ai-use-cases

Will Knight. "AI Can Write Disinformation Now—and Dupe Human Readers." [ออนไลน์] 2021. [สืบค้นวันที่ 7 ธันวาคม 2566]. จาก https://www.wired.com/story/ai-write-disinformation-dupe-human-readers/.

S. Kreps, R. M. McCain, and M. Brundage, “All the News That’s Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation,” Journal of Experimental Political Science, vol. 9, no. 1, pp. 104–117, 2022. https://doi.org/10.1017/XPS.2020.37

Inc. GitHub. "GitHub Copilot Your AI pair programmer." [ออนไลน์] 2023. [สืบค้นวันที่ 7 ธันวาคม 2566]. จาก https://github.com/features/copilot.

Vasconcelos, Helena and Bansal, Gagan and Fourney, Adam and Liao, Q. Vera and Vaughan, Jennifer Wortman, “Generation Probabilities Are Not Enough: Uncertainty Highlighting in AI Code Completions” ACM Trans. Comput.-Hum. Interact., Just Accepted, 2024. https://doi.org/10.1145/3702320.

Linkedin. "Top 10 Generative AI Tools and Platforms of 2024" [ออนไลน์] 2024. [สืบค้นวันที่ 3 เมษายน 2568]. จาก https://www.linkedin.com/pulse/top-10-generative-ai-tools-platforms-2024-yjmtc.

R. Khoury, A. R. Avila, J. Brunelle and B. M. Camara, "How Secure is Code Generated by ChatGPT?," 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, Oahu, HI, USA, 2023, pp. 2445-2451, doi: 10.1109/SMC53992.2023.10394237.

D. Horne, "PwnPilot: Reflections on Trusting Trust in the Age of Large Language Models and AI Code Assistants," 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 2023, pp. 2457-2464, doi: 10.1109/CSCE60160.2023.00396.

Tiago Espinha Gasiba, Kaan Oguzhan, Ibrahim Kessba, Ulrike Lechner, and Maria Pinto-Albuquerque. I'm Sorry Dave, I'm Afraid I Can't Fix Your Code: On ChatGPT, CyberSecurity, and Secure Coding. In 4th International Computer Programming Education Conference (ICPEC 2023). Open Access Series in

Informatics (OASIcs), Volume 112, pp. 2:1-2:12, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2023) https://doi.org/10.4230/OASIcs.ICPEC.2023.2

Mahesh Jamdade and Yi Liu. 2024. A Pilot Study on Secure Code Generation with ChatGPT for Web Applications. In Proceedings of the 2024 ACM Southeast Conference (ACMSE '24). Association for Computing Machinery, New York, NY, USA, 229–234. https://doi.org/10.1145/3603287.3651194

Yamagishi, R., Sasa, S. and Fujii, S., Analysis of the Effect of the Difference between Japanese and English Input on ChatGPT-Generated Secure Codes. https://www.ndss-symposium.org/wp-content/uploads/madweb2024-84-paper.pdf

Suto, Larry. "Analyzing the accuracy and time costs of web application security scanners." San Francisco, February 1 (2010). https://www.think-secure.nl/uk/Accuracy_and_Time_Costs_of_Web_App_Scanners.pdf

Althunayyan M, Saxena N, Li S, Gope P. Evaluation of Black-Box Web Application Security Scanners in Detecting Injection Vulnerabilities. Electronics. 2022; 11(13):2049. https://doi.org/10.3390/electronics11132049

Zenah, Nor & Aziz, N.A.. (2011). Secure coding in software development. 2011 5th Malaysian Conference in Software Engineering, MySEC 2011. 458-464. 10.1109/MySEC.2011.6140716.

Yin Z, Lee SU-J. Security Analysis of Web Open-Source Projects Based on Java and PHP. Electronics. 2023; 12(12):2618. https://doi.org/10.3390/electronics12122618

XAMPP. What is XAMPP? [ออนไลน์] 2567. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://www.apachefriends.org/index.html

Apache. What is the Apache HTTP Server Project? [ออนไลน์] 1997-2025. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://httpd.apache.org/ABOUT_APACHE.html

Mell, Peter, Karen Scarfone, and Sasha Romanosky. "A complete guide to the common vulnerability scoring system version 2.0." Published by FIRST-forum of incident response and security teams. Vol. 1. 2007.

Cloudflare. What is SQL injection (SQi)? [ออนไลน์] 2568. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://www.cloudflare.com/learning/security/threats/sql-injection/

OWASP. Cross Site Scripting (XSS) [ออนไลน์] 2568. [สืบค้นวันที่ 21 มกราคม 2568]. จาก https://owasp.org/www-community/attacks/xss/

ภาคผนวก
ข้อมูลการประเมินช่องโหว่

ผลการประเมินช่องโหว่ของแต่ละรอบที่ทำการสแกนด้วย OWASP ZAP

ตารางที่ ก-1 ผลการประเมินช่องโหว่ของระบบหน้าเว็บเพจ รอบที่ 1

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Absence of Anti-CSRF Tokens	A01	352	✔	✔	✔
2	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
3	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
4	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
5	Strict-Transport-Security Header Not Set	A05	319	✔	✔	✔
6	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
7	Authentication Request Identified	-	-	✔	✔	✔
8	GET for POST	A04	16	✔	✔	✔
9	Re-examine Cache-control Directives	-	525	✔	✔	✔
10	User Agent Fuzzer	-	-	✔	✔	✔

ตารางที่ ก-1 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
11	Modern Web Application	-	-			✔

ตารางที่ ก-2 ผลการประเมินช่องโหว่ของระบบหน้าเว็บเพจ รอบที่ 2

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Absence of Anti-CSRF Tokens	A01	352		✔	
2	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
3	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
4	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
5	Strict-Transport-Security Header Not Set	A05	319	✔	✔	✔
6	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
7	Authentication Request Identified	-	-	✔	✔	✔
8	GET for POST	A04	16	✔	✔	
9	Re-examine Cache-control Directives	-	525	✔	✔	✔
10	User Agent Fuzzer	-	-	✔	✔	✔

ตารางที่ ก-2 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
11	Modern Web Application	-	-			
12	User Controllable HTML Element Attribute (Potential XSS)	-	-	✔		
13	CSP: Wildcard Directive	A05	693		✔	✔
14	Information Disclosure - Suspicious Comments	A01	200			✔
15	CSP: style-src unsafe-inline	A05	693			✔
16	X-Frame-Options Defined via META (Non-compliant with Spec)	A05	1021			✔

ตารางที่ ก-3 ผลการประเมินช่องโหว่ของระบบหน้าเว็บเพจ รอบที่ 3

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Absence of Anti-CSRF Tokens	A01	352			
2	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
3	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
4	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
5	Strict-Transport-Security Header Not Set	A05	319	✔	✔	✔

ตารางที่ ก-3 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
6	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
7	Authentication Request Identified	-	-	✔	✔	✔
8	GET for POST	A04	16	✔	✔	
9	Re-examine Cache-control Directives	-	525	✔	✔	✔
10	User Agent Fuzzer	-	-	✔	✔	✔
11	Modern Web Application	-	-			
12	User Controllable HTML Element Attribute (Potential XSS)	-	-	✔	✔	
13	CSP: Wildcard Directive	A05	693	✔	✔	
14	Information Disclosure - Suspicious Comments	A01	200			✔
15	CSP: style-src unsafe-inline	A05	693	✔		
16	X-Frame-Options Defined via META (Non-compliant with Spec)	A05	1021	✔		✔
17	Strict-Transport-Security Defined via META (Non-compliant with Spec)	A05	319	✔		✔
18	CSP: Meta Policy Invalid Directive	A05	693	✔		✔

ตารางที่ ก-4 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันระบบตรวจสอบตัวตน รอบที่ 1

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	SQL Injection - MySQL	A03	89	✔	✔	✔
2	Absence of Anti-CSRF Tokens	A01	352	✔	✔	✔
3	Application Error Disclosure	A01	200			
4	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
5	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
6	Parameter Tampering	A04	472			✔
7	Cookie No HttpOnly Flag	A05	1004	✔	✔	✔
8	Cookie Without Secure Flag	A05	614	✔	✔	
9	Cookie without SameSite Attribute	A01	1275	✔	✔	✔
10	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200	✔	✔	✔
11	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
12	Strict-Transport-Security Header Not Set	A05	319	✔	✔	
13	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔

ตารางที่ ก-4 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
14	Authentication Request Identified	-	-	✔	✔	✔
15	GET for POST	A04	16			✔
16	Re-examine Cache-control Directives	-	525	✔	✔	
17	Session Management Response Identified	-	-	✔	✔	✔
18	User Agent Fuzzer	-	-			✔

ตารางที่ ก-5 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันระบบตรวจสอบตัวตน รอบที่ 2

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	SQL Injection - MySQL	A03	89			
2	Absence of Anti-CSRF Tokens	A01	352		✔	✔
3	Application Error Disclosure	A01	200			
4	CSP: Wildcard Directive	A05	693	✔		
5	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
6	Missing Anti-clickjacking Header	A05	1021		✔	✔
7	Parameter Tampering	A04	472		✔	✔
8	Cookie No HttpOnly Flag	A05	1004	✔		✔

ตารางที่ ก-5 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
9	Cookie Without Secure Flag	A05	614	✔		
10	Cookie without SameSite Attribute	A01	1275	✔		✔
11	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200	✔	✔	✔
12	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
13	Strict-Transport-Security Header Not Set	A05	319	✔	✔	
14	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
15	Authentication Request Identified	-	-	✔	✔	✔
16	GET for POST	A04	16	✔	✔	
17	Re-examine Cache-control Directives	-	525		✔	
18	Session Management Response Identified	-	-	✔		✔
19	User Agent Fuzzer	-	-	✔	✔	✔

ตารางที่ ก-6 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันระบบตรวจสอบตัวตน รอบที่ 3

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	SQL Injection - MySQL	A03	89			
2	Absence of Anti-CSRF Tokens	A01	352			
3	Application Error Disclosure	A01	200	✔		
4	CSP: Wildcard Directive	A05	693	✔		✔
5	CSP: script-src unsafe-inline	A05	693			✔
6	CSP: style-src unsafe-inline	A05	693			✔
7	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
8	Missing Anti-clickjacking Header	A05	1021			
9	Parameter Tampering	A04	472		✔	
10	Cookie No HttpOnly Flag	A05	1004			✔
11	Cookie Without Secure Flag	A05	614			
12	Cookie without SameSite Attribute	A01	1275			✔
13	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200	✔	✔	✔

ตารางที่ ก-6 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
14	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
15	Strict-Transport-Security Header Not Set	A05	319			
16	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
17	Authentication Request Identified	-	-	✔	✔	✔
18	GET for POST	A04	16			
19	Re-examine Cache-control Directives	-	525			
20	Session Management Response Identified	-	-	✔		✔
21	User Agent Fuzzer	-	-	✔	✔	✔
22	User Controllable HTML Element Attribute (Potential XSS)	-	-			✔

ตารางที่ ก-7 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันจัดการไฟล์ รอบที่ 1

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Cross Site Scripting (Persistent)				✔	
2	SQL Injection - MySQL	A03	89	✔	✔	

ตารางที่ ก-7 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
3	Absence of Anti-CSRF Tokens	A01	352	✔	✔	
4	Application Error Disclosure	A01	200	✔	✔	✔
5	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
6	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
7	Parameter Tampering	A04	472	✔	✔	✔
8	Big Redirect Detected (Potential Sensitive Information Leak)	A04	201	✔	✔	
9	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200	✔	✔	✔
10	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
11	Strict-Transport-Security Header Not Set	A05	319		✔	
12	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
13	Re-examine Cache-control Directives	-	525		✔	
14	User Agent Fuzzer	-	-	✔	✔	✔

ตารางที่ ก-7 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
15	User Controllable HTML Element Attribute (Potential XSS)	-	-		✔	

ตารางที่ ก-8 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันจัดการไฟล์ รอบที่ 2

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Cross Site Scripting (Persistent)	A03	79			
2	SQL Injection - MySQL	A03	89		✔	
3	Absence of Anti-CSRF Tokens	A01	352	✔	✔	
4	Application Error Disclosure	A01	200	✔	✔	
5	CSP: Wildcard Directive	A05	693			✔
6	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
7	Missing Anti-clickjacking Header	A05	1021	✔	✔	✔
8	Parameter Tampering	A04	472	✔	✔	
9	Big Redirect Detected (Potential Sensitive Information Leak)	A04	201	✔		
10	Cookie No HttpOnly Flag	A05	1004			✔

ตารางที่ ก-8 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
11	Cookie without SameSite Attribute	A01	1275			✔
12	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200	✔	✔	✔
13	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
14	Strict-Transport-Security Header Not Set	A05	319			
15	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
16	Re-examine Cache-control Directives	-	525			
17	Session Management Response Identified	-	-			✔
18	User Agent Fuzzer	-	-	✔	✔	✔
19	User Controllable HTML Element Attribute (Potential XSS)	-	-		✔	

ตารางที่ ก-9 ผลการประเมินช่องโหว่เว็บแอปพลิเคชันจัดการไฟล์ รอบที่ 3

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
1	Cross Site Scripting (Persistent)	A03	79			

ตารางที่ ก-9 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
2	SQL Injection - MySQL	A03	89			
3	Absence of Anti-CSRF Tokens	A01	352			
4	Application Error Disclosure	A01	200			
5	CSP: Wildcard Directive	A05	693	✔	✔	✔
6	Content Security Policy (CSP) Header Not Set	A05	693	✔	✔	✔
7	Missing Anti-clickjacking Header	A05	1021			
8	Parameter Tampering	A04	472	✔	✔	
9	Big Redirect Detected (Potential Sensitive Information Leak)	A04	201			
10	Cookie No HttpOnly Flag	A05	1004			✔
11	Cookie without SameSite Attribute	A01	1275			✔
12	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	A01	200			✔
13	Server Leaks Version Information via "Server" HTTP Response Header Field	A05	200	✔	✔	✔
14	Strict-Transport-Security Header Not Set	A05	319			

ตารางที่ ก-9 (ต่อ)

ลำดับ	รายการช่องโหว่	OWASP	CWE	ChatGPT	Gemini	Claude
15	X-Content-Type-Options Header Missing	A05	693	✔	✔	✔
16	Re-examine Cache-control Directives	-	525			
17	GET for POST	A04	16	✔	✔	✔
18	Session Management Response Identified	-	-	✔	✔	✔
19	User Agent Fuzzer	-	-	✔	✔	✔
20	User Controllable HTML Element Attribute (Potential XSS)	-	-			

ประวัติผู้เขียน

ชื่อ	อธิวัฒน์ สุภอมรพันธุ์
ชื่อการค้นคว้าอิสระ	การทดสอบประสิทธิภาพของ Generative AI ในการสร้างและแก้ไขช่องโหว่บนเว็บแอปพลิเคชัน
สาขาวิชา	การบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ประวัติ	เกิดเมื่อวันที่ 17 กรกฎาคม 2537 ปัจจุบันอายุ 30 ปี โดยเกิดที่จังหวัดกรุงเทพมหานคร เข้าศึกษาระดับปริญญาตรี ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีและการจัดการอุตสาหกรรม ที่มหาวิทยาลัยพระจอมเกล้าพระนครเหนือจนสำเร็จการศึกษาในปีการศึกษา 2559 และเข้าศึกษาต่อระดับปริญญาโท สาขาการบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในด้านการทำงาน มีประสบการณ์ดังต่อไปนี้ • พ.ศ. 2559 – 2561 วิศวกรเครือข่าย บริษัท แอ็ดวานซ์อินฟอร์เมชั่นเทคโนโลยี จำกัด (มหาชน) • พ.ศ. 2561 – 2564 วิศวกรเครือข่าย บริษัท ดาต้าโปรคอมพิวเตอร์ ซิสเต็มส์ จำกัด • พ.ศ. 2564 ถึงปัจจุบัน วิศวกรเครือข่าย บริษัท เอ็นทีที เดต้า (ประเทศไทย) จำกัด