ระบบตรวจสอบความปลอดภัยของซอร์สโค้ด CodeGuard

นายวรเมศ สุขเจริญ[1],กิตติกร หาญตระกูล[1] ,ปวีณ เขื่อนแก้ว[1] และ สมนึก สินธุปวน[1*]

Woramate Sukcharoen[1], Kittikorn Hantrakul[1] , Paween Khoenkaw [1] and Somneuk Sintupuan [1*]

[1] *สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้เชียงใหม่ 50290 ประเทศไทย*

* ผู้นิพนธ์ประสานงาน: กิตติกร หาญตระกูล อีเมล: kittikorn@gmaejo.mju.ac.th

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์ในการออกแบบและพัฒนาเครื่องมือวิเคราะห์ความปลอดภัยของซอร์สโค้ด (Static Application Security Testing: SAST) ภายใต้ชื่อระบบ CodeGuard เพื่อช่วยให้นักพัฒนาซอฟต์แวร์และนักศึกษาสามารถตรวจสอบช่องโหว่ด้านความมั่นคงปลอดภัยในซอร์สโค้ดได้ด้วยตนเองบนเครื่องของตน โดยไม่ต้องส่งโค้ดขึ้นสู่เซิร์ฟเวอร์ภายนอก

ระบบที่พัฒนาขึ้นรองรับการสแกนซอร์สโค้ดจากโฟลเดอร์ ไฟล์เดี่ยว และคลังซอร์สโค้ดบน GitHub ใช้เทคนิคการวิเคราะห์ทั้งแบบอาศัยรูปแบบข้อความ (Regex-based) และการวิเคราะห์โครงสร้างโค้ดด้วยโครงสร้างรูปแบบต้นไม้ (Abstract Syntax Tree : AST) พร้อมทั้งออกแบบระบบกฎ (Rules) ในรูปแบบไฟล์ YAML ที่จัดกลุ่มตามแนวคิดของ OWASP Top 10 และรองรับหลายภาษาคอมพิวเตอร์ผ่านโครงสร้างปลั๊กอินภาษา นอกจากนี้ยังมีส่วนติดต่อผู้ใช้แบบกราฟิก (GUI) ที่แสดงผลการสแกนในรูปแบบแดชบอร์ด (Dashboard) รายการประวัติการสแกน และหน้ารายละเอียดช่องโหว่ รวมถึงสามารถส่งออกรายงานในรูปแบบไฟล์ HTML ได้

ผลจากการทดลองใช้งานกับชุดโค้ดตัวอย่างและโปรเจกค์ขนาดเล็กพบว่า CodeGuard สามารถตรวจพบรูปแบบโค้ดที่มีความเสี่ยงตามกฎที่กำหนด และแสดงผลในรูปแบบที่ผู้ใช้เข้าใจได้ง่าย โดยรวมแล้วระบบที่พัฒนาขึ้นช่วยให้การตรวจสอบความมั่นคงปลอดภัยเบื้องต้นของซอร์สโค้ดมีความสะดวก รวดเร็ว และเหมาะสมต่อการนำไปใช้สนับสนุนการทำงานและการเรียนการสอนด้านวิทยาการคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

คำสำคัญ: เครื่องมือวิเคราะห์ความปลอดภัยของซอร์สโค้ด, การทดสอบความปลอดภัยแบบสถิตย์ (SAST), OWASP Top 10, การวิเคราะห์ซอร์สโค้ดด้วย Regex และ AST, CodeGuard

Abstract

This project aims to design and develop a source code security analysis tool (Static Application Security Testing: SAST) called CodeGuard to enable software developers and students to detect security vulnerabilities in source code locally on their own machines, without uploading code to external servers.

The proposed system supports scanning source code from folders, single files, and GitHub repositories. It employs both regex-based analysis and Abstract Syntax Tree (AST)–based structural analysis, and introduces a rule system defined in YAML files, organized according to the OWASP Top 10. The tool supports multiple programming languages through a plugin-based architecture, and

provides a graphical user interface (GUI) that presents scan results in the form of a dashboard, scan history, and detailed vulnerability views, as well as the ability to export reports in HTML format.

Experiments using sample code and small projects show that CodeGuard can detect risky code patterns according to the defined rules and present the results in a way that is easy for users to understand. Overall, the developed system makes preliminary source code security assessment more convenient and faster, and is suitable for supporting both practical work and computer science education effectively.

Keywords: Static Application Security Testing (SAST), source code security analysis, OWASP Top 10, regex-based analysis, Abstract Syntax Tree (AST), plugin-based architecture, CodeGuard