การประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์

นายวุฒิภัทร พุ่มพวง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์

นายวุฒิภัทร พุ่มพวง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์

โดย นายวุฒิภัทร พุ่มพวง

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 : นายวุฒิภัทร พุ่มพวง
ชื่อการค้นคว้าอิสระ : การประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์
สาขาวิชา : การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์
ปีการศึกษา : 2567

บทคัดย่อ

การวิจัยนี้มุ่งเน้นการประเมินความมั่นคงปลอดภัยและการทดสอบช่องโหว่ของเว็บไซต์ Moodle โดยใช้เวอร์ชัน 3.7.1 และ 4.5.1 ซึ่งติดตั้งผ่าน Docker บนระบบปฏิบัติการ Ubuntu 24.04.1 LTS และจัดการด้วย Proxmox เพื่อให้สามารถจำลองสภาพแวดล้อมการทดสอบได้อย่างยืดหยุ่น การใช้ Docker ช่วยลดความซับซ้อนของการบริหารจัดการโครงสร้างพื้นฐาน และช่วยให้สามารถเปรียบเทียบผลการทดสอบระหว่างสองเวอร์ชันได้อย่างมีประสิทธิภาพการศึกษานี้ใช้เครื่องมือโอเพนซอร์สด้านความมั่นคงปลอดภัย ได้แก่ OWASP ZAP, OpenVAS และ Nessus ในการวิเคราะห์และตรวจจับช่องโหว่สำคัญของเว็บไซต์ Moodle โดยเน้นการประเมินความมั่นคงปลอดภัยที่เกี่ยวข้องกับ SQL Injection, Cross-Site Scripting (XSS), Remote Code Execution (RCE) และ ช่องโหว่ด้านการกำหนดค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์

เป้าหมายของการวิจัยนี้คือการระบุช่องโหว่ที่ส่งผลกระทบต่อความมั่นคงปลอดภัยของเว็บไซต์ Moodle และ เปรียบเทียบผลการทดสอบ ระหว่างเวอร์ชัน 3.7.1 และ 4.5.1 เพื่อประเมินพัฒนาการในการปรับปรุงความมั่นคงปลอดภัยของระบบ รวมถึงการพัฒนา แนวทางปฏิบัติที่ดีที่สุด (Best Practices) สำหรับการเพิ่มความมั่นคงปลอดภัยของ Moodle โดยใช้มาตรฐาน CVSS (Common Vulnerability Scoring System) ในการวิเคราะห์ระดับความรุนแรงของช่องโหว่ที่พบและเสนอแนวทางแก้ไข เช่น การตั้งค่า SSL/TLS การจัดการสิทธิ์ผู้ใช้ (RBAC) และการกำหนด Content Security Policy (CSP) เพื่อเพิ่มระดับการป้องกันระบบ

ผลการวิจัยคาดว่าจะช่วยเสริมสร้างความมั่นคงปลอดภัยและเพิ่มความน่าเชื่อถือให้กับเว็บไซต์ Moodle ของวิทยาลัยการศึกษาที่มีทรัพยากรจำกัด ลดความเสี่ยงจากการโจมตีทางไซเบอร์ที่อาจเกิดขึ้นในอนาคต รวมถึงช่วยให้สามารถใช้ เครื่องมือโอเพนซอร์สด้านความมั่นคงปลอดภัย ในการจัดการและตรวจสอบช่องโหว่ได้อย่างมีประสิทธิภาพมากขึ้น

(มีจำนวนทั้งสิ้น 86 หน้า)

คำสำคัญ : มูเดิล ช่องโหว่ การฝังคำสั่ง SQL การฝังสคริปต์ข้ามไซต์ เครื่องมือ OWASP ZAP เครื่องมือ OpenVAS เครื่องมือ Nessus

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Wutthiphat Phumphuang
Independent Study Title : Vulnerability Assessment of online learning platform
Major Field : Network and Information Security Management
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Nawaporn Wisitpongphan
Academic Year : 2024

ABSTRACT

This research focuses on the security assessment and vulnerability testing of Moodle websites, specifically versions 3.7.1 and 4.5.1, which are deployed via Docker on the Ubuntu 24.04.1 LTS operating system and managed using Proxmox. This setup allows for a flexible testing environment simulation. The use of Docker simplifies infrastructure management and enables efficient comparison of security assessments between the two versions. The study utilizes open-source security tools, including OWASP ZAP, OpenVAS, and Nessus, to analyze and detect critical vulnerabilities in Moodle. The security evaluation primarily focuses on SQL Injection, Cross-Site Scripting (XSS), Remote Code Execution (RCE), and server security misconfigurations.

The objective of this research is to identify vulnerabilities affecting Moodle's security and to compare the test results between versions 3.7.1 and 4.5.1 to assess improvements in system security. Additionally, it aims to develop best practices for enhancing Moodle's security by applying the Common Vulnerability Scoring System (CVSS) to analyze vulnerability severity levels and propose mitigation strategies such as SSL/TLS configuration, role-based access control (RBAC), and Content Security Policy (CSP) enforcement to strengthen system protection.

The expected outcome of this study is to enhance security and reliability for Moodle websites used by educational institutions with limited resources, reducing the risk of potential cyberattacks. Moreover, this research aims to promote the effective use of open-source security tools for managing and assessing vulnerabilities efficiently.

(Total 86 Pages)

Keywords: Moodle, Vulnerabilities, SQL Injection, XSS, OWASP ZAP, OpenVAS, Nessus

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระในครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความร่วมมือ และความช่วยเหลือ จากบุคคลหลายท่าน ที่ให้คำปรึกษาและความช่วยเหลือในด้านต่าง ๆ อันเป็นประโยชน์ ต่อการดำเนินงานการค้นคว้าอิสระ ขอกราบขอบพระคุณที่ปรึกษาสารนิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. นวพร วิสิฐพงศ์พันธ์ และคณะกรรมการทุกท่านที่ได้ให้คำปรึกษา ตรวจสอบ แก้ไข คำแนะนำ และข้อเสนอแนะที่เป็นแนวทางในการจัดทำสารนิพนธ์ให้สำเร็จได้

สุดท้ายนี้ผู้วิจัยขอขอบคุณอาจารย์ประจำคณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล ทุก ท่านที่ได้ให้ความรู้ คำแนะนำตลอดการศึกษา และขอขอบคุณเพื่อนนักศึกษา สาขาวิชาการบริหาร เครือข่าย และความมั่นคงปลอดภัยสารสนเทศ คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัลมหา วิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ที่คอยช่วยเหลือกันและกันในทุก ๆ เรื่อง

ท้ายที่สุดนี้ ผู้วิจัยขอกราบขอบพระคุณครอบครัว และเพื่อน ๆ พี่ ๆ น้อง ๆ ทุกท่าน ที่ให้การ สนับสนุนและเป็นกำลังใจให้เสมอมาจนการค้นคว้าอิสระนี้สำเร็จลุล่วงไปได้ด้วยดี

วุฒิภัทร พุ่มพวง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูปภาพ

ภาพที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคดิจิทัลปัจจุบัน การใช้ระบบการจัดการการเรียนรู้ (Learning Management System หรือ LMS) ได้กลายเป็นสิ่งจำเป็นสำหรับวิทยาลัยการศึกษาทุกระดับเพื่อช่วยอำนวยความสะดวกในการจัดการการเรียนการสอนออนไลน์ Moodle เป็นหนึ่งในระบบ LMS ที่ได้รับความนิยมอย่าง แพร่หลายในหลายวิทยาลัยการศึกษาเนื่องจากเป็นโอเพนซอร์ส (Open Source) และมีความยืดหยุ่น ในการปรับแต่งเพื่อให้ตรงกับความต้องการของผู้ใช้ [1] ในการพัฒนาสภาพแวดล้อมที่เหมาะสม สำหรับการทดสอบ Moodle ระบบ Docker ถูกเลือกใช้เนื่องจากความสามารถในการจำลองระบบ ที่มีความยืดหยุ่นและสามารถปรับแต่งได้ การใช้ Ubuntu ซึ่งเป็นระบบปฏิบัติการที่ได้รับความนิยม และได้รับการสนับสนุนระยะยาวช่วยสร้างความเสถียรในระบบ และประสิทธิภาพในการจำลองระบบ อย่างไรก็ตามการใช้งาน Moodle โดยเฉพาะในวิทยาลัยขนาดเล็กมักประสบปัญหาด้านการรักษา ความปลอดภัยของเว็บไซต์ เนื่องจากขาดทรัพยากรในการพัฒนาระบบป้องกันที่เข้มงวด

วิทยาลัยขนาดเล็กมักจะมีข้อจำกัดทางด้านงบประมาณและทรัพยากรบุคคลทำให้ไม่สามารถลงทุนในระบบความมั่นคงปลอดภัยทางไซเบอร์ที่ทันสมัยได้ เช่น การสแกนช่องโหว่ของเว็บไซต์ที่ใช้ หรือ การจ้างบุคลากรที่มีความรู้ความเชี่ยวชาญทางด้านความมั่นคงปลอดภัย นอกจากนี้การขาดความตระหนักรู้ เกี่ยวกับการโจมตีทางไซเบอร์ และการตั้งค่าความมั่นคงปลอดภัยที่ไม่เหมาะสม ยังเป็นปัจจัยเสี่ยงที่ทำให้ เว็บไซต์ของวิทยาลัยเป็นเป้าหมายที่ง่ายต่อการโจมตีจากแฮ็กเกอร์

จากข้อมูลของรายงานโดย Cybersecurity Ventures [2] ได้มีการระบุว่าการโจมตีที่เกิดขึ้นกับหน่วยงานวิทยาลัยการศึกษาเพิ่มขึ้นอย่างมากในช่วงไม่กี่ปีที่ผ่านมา โดยเฉพาะในวิทยาลัยและมหาวิทยาลัยที่มีขนาดเล็กที่มีงบประมาณจำกัดในการลงทุนด้านความมั่นคงปลอดภัยทางไซเบอร์ ซึ่งทำให้การใช้งาน Docker และ Ubuntu ช่วยลดต้นทุน และทรัพยากรในการจัดการสภาพแวดล้อมสามารถทำงานในระบบที่ปรับปรุง และสร้างซ้ำได้ง่ายโดยไม่ต้องพึ่งพาโครงสร้างพื้นฐานที่ซับซ้อนหรือมีค่าใช้จ่ายสูงซึ่งเหมาะสมสำหรับวิทยาลัยขนาดเล็กที่มีข้อจำกัดด้านทรัพยากรนอกจากนี้งานวิจัยโดย EDUCAUSE [3] ยังชี้ให้เห็นว่าการโจมตีทางไซเบอร์ที่มุ่งเป้าไปยังเว็บไซต์ต่าง ๆ ของ วิทยาลัยการศึกษามักเกี่ยวข้องกับการขโมยข้อมูลผู้ใช้งาน การฝังมัลแวร์ที่หลากหลายรูปแบบ และการเข้าถึงข้อมูลที่เป็นความลับของนักศึกษา และบุคลากรซึ่งทั้งหมดนี้มีผลกระทบต่อชื่อเสียงของวิทยาลัย และความน่าเชื่อถือของวิทยาลัยโดยตรง

Sandhu และคณะ [8] เน้นความสำคัญของการการกำหนดสิทธิ์การเข้าถึงตามบทบาท (Role-based Access Control - RBAC) ของผู้ใช้งาน เพื่อเป็นแนวทางที่ช่วยลดความเสี่ยงในการถูกโจมตีจากการเข้าถึงที่ไม่ได้รับอนุญาต ประกอบกับการใช้งาน SSL/TLS เพื่อเข้ารหัสข้อมูลระหว่างการรับ

และส่งข้อมูลที่สำคัญ เช่น ข้อมูลส่วนบุคคล และรหัสผ่านของผู้ใช้งาน ซึ่งการตั้งค่าเหล่านี้เป็นพื้นฐานในการพัฒนาเว็บไซต์ Moodle ให้มีความมั่นคงปลอดภัย [7]

ดังนั้น การทำวิจัยเพื่อศึกษาช่องโหว่และแนวทางการปรับปรุงความมั่นคงปลอดภัยสำหรับเว็บไซต์ Moodle บนสภาพแวดล้อมด้วย Docker และ Ubuntu จะทำให้สามารถนำเสนอ Best Practices เพื่อนำไปปรับใช้ได้อย่างสะดวกในวิทยาลัยที่มีทรัพยากรจำกัด การศึกษาในครั้งนี้ไม่เพียงแต่จะช่วยให้วิทยาลัยเข้าใจถึงความเสี่ยงและความเปราะบางที่อาจเกิดขึ้น แต่ยังช่วยพัฒนามาตรการป้องกันที่เหมาะสมและปกป้องข้อมูลที่สำคัญของวิทยาลัยได้อย่างมีประสิทธิภาพ มากขึ้น

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อประเมินระดับความมั่นคงปลอดภัยของเว็บไซต์ Moodle ที่ใช้ในวิทยาลัยการศึกษา โดยมุ่งเน้นไปที่การระบุช่องโหว่ด้านความมั่นคงปลอดภัยที่อาจเกิดขึ้น และเสนอแนวทางในการแก้ไขปัญหา เพื่อให้ระบบมีความมั่นคงปลอดภัยมากยิ่งขึ้น สามารถปกป้องข้อมูลที่สำคัญของวิทยาลัย และผู้ใช้งานได้อย่างมีประสิทธิภาพ

1.2.2 เพื่อจัดทำแนวทางการพัฒนาเว็บไซต์โดยใช้ Moodle ที่มีความมั่นคงปลอดภัย เพื่อพัฒนา กระบวนการสร้างและปรับปรุงเว็บไซต์ที่ใช้ Moodle โดยเน้นการรักษาความมั่นคงปลอดภัย ของข้อมูลและการเพิ่มประสิทธิภาพในการป้องกันการโจมตีทางไซเบอร์ในวิทยาลัยการศึกษา

1.2.3 เพื่อเปรียบเทียบผลการทดสอบช่องโหว่ระหว่างเวอร์ชัน Moodle 3.7.1 และ Moodle 4.5.1 เพื่อตรวจสอบและแสดงให้เห็นถึงความก้าวหน้าในการปรับปรุงความมั่นคงปลอดภัยของระบบรวมถึงนำผลที่ได้จัดทำข้อเสนอแนะที่สามารถนำไปปฏิบัติได้จริงในวิทยาลัยการศึกษาที่มีทรัพยากรจำกัด

1.3 ขอบเขตของการวิจัย

1.3.1 เครื่องมือที่ใช้

1.3.1.1 Docker และ Ubuntu ใช้สำหรับจำลองสภาพแวดล้อมการทดสอบ และติดตั้ง Moodle เวอร์ชัน 3.7.1 และ 4.5.1 โดย Docker ช่วยเพิ่มความยืดหยุ่นในการตั้งค่า และจัดการสภาพแวดล้อมการทดสอบ พร้อมทั้งลดความยุ่งยากในการติดตั้งซอฟต์แวร์

1.3.1.2 การจำลองระบบปฏิบัติการ Linux และติดตั้ง Moodle บน Proxmox หรือ VMWare

1.3.1.3 HP ProLiant DL120 G7 quad core Intel Xeon E31240 CPU 2.868 GHz ใช้เป็นเซิร์ฟเวอร์สำหรับการจำลองระบบ

1.3.1.4 DELL Precision 5530 Intel(R) Core i7-8850H CPU 2.60GHz ใช้เป็นเครื่องสำหรับการสแกน และตรวจหาช่องโหว่

1.3.1.5 OWASP ZAP ใช้สำหรับการสแกน และตรวจหาช่องโหว่ในเว็บแอปพลิเคชัน Moodle โดยเน้นการทดสอบช่องโหว่ทั่วไป เช่น SQL Injection, Cross-Site Scripting (XSS) และ Remote Code Execution (RCE)

1.3.1.6 OpenVAS เป็นเครื่องมือสำหรับ การสแกนและประเมินช่องโหว่ด้านความมั่นคงปลอดภัยของระบบเครือข่ายและเซิร์ฟเวอร์ สามารถตรวจจับช่องโหว่ที่เกี่ยวข้องกับ การตั้งค่าซอฟต์แวร์ ระบบปฏิบัติการ และบริการเครือข่าย รวมถึงให้คำแนะนำแนวทางแก้ไขตามมาตรฐาน CVSS

1.3.1.7 Nessus เป็นเครื่องมือสำหรับ การสแกนช่องโหว่ในระบบเครือข่ายและเว็บแอปพลิเคชัน โดยสามารถตรวจจับช่องโหว่ที่ทราบอยู่แล้ว (Known Vulnerabilities) ตามฐานข้อมูล CVE พร้อมให้คำแนะนำแนวทางแก้ไข รองรับการตรวจสอบ SSL/TLS Configuration การตั้งค่าผิดพลาดของเซิร์ฟเวอร์ และความมั่นคงปลอดภัยของ Moodle SQLMap ใช้สำหรับตรวจสอบและโจมตี SQL Injection โดยอัตโนมัติ บนเว็บไซต์ Moodle เพื่อหาช่องโหว่ที่เกี่ยวข้องกับฐานข้อมูล

1.3.2 ระบบที่ใช้ในการทดสอบประกอบด้วยแพลตฟอร์ม Moodle 3.7.1 และ Moodle 4.5.1 ซึ่งติดตั้งบน Ubuntu 24.04.1 LTS ผ่าน Docker และบริหารจัดการโดย Proxmox เพื่อให้สามารถควบคุมสภาพแวดล้อมการทดสอบได้อย่างมีประสิทธิภาพ ระบบใช้ MariaDB เป็นฐานข้อมูล และให้บริการผ่าน Apache/Nginx เป็นเว็บเซิร์ฟเวอร์ ทั้งนี้ การประเมินความมั่นคงปลอดภัยดำเนินการโดยใช้เครื่องมือ OWASP ZAP, OpenVAS และ Nessus เพื่อตรวจจับและวิเคราะห์ช่องโหว่ของระบบ

1.3.3 การทดสอบช่องโหว่ของระบบ Moodle 3.7.1 และ 4.5.1 มุ่งเน้นไปที่การจัดการอินพุต (Input Validation) เช่น SQL Injection (SQLi) Cross-Site Scripting (XSS) และ Remote Code Execution (RCE) การยืนยันตัวตนและการควบคุมสิทธิ์ (Authentication & Access Control) เช่น Brute Force Attack Session Management และ Privilege Escalation การตั้งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ (Server Security Configuration) เช่น HTTP Security Headers, SSL/TLS Configuration และ Directory Listing ช่องโหว่จากปลั๊กอิน และการจัดการไฟล์ (Plugin & File Security) เช่น การอัปโหลดไฟล์ที่ไม่ปลอดภัย และการใช้ปลั๊กอินที่ล้าสมัย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 เพิ่มความมั่นคงปลอดภัยให้กับเว็บไซต์ของวิทยาลัย

1.4.2 ประหยัดค่าใช้จ่ายในการรักษาความมั่นคงปลอดภัย โดยวิทยาลัยจะมีแนวทางการนำเครื่องมือโอเพนซอร์ส เช่น OWASP ZAP OpenVAS และ Nessus มาใช้เพื่อลดต้นทุนในการจัดหาซอฟต์แวร์ความมั่นคงปลอดภัยโดยไม่จำเป็นต้องใช้ซอฟต์แวร์ ที่มีค่าใช้จ่ายสูง

1.4.3 เสริมสร้างความน่าเชื่อถือของระบบการเรียนการสอนออนไลน์ของวิทยาลัย

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

การวิจัยครั้งนี้ ผู้วิจัยได้ศึกษาค้นคว้าทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อสนับสนุนการดำเนินการวิจัยให้สามารถออกแบบวิเคราะห์ และทดสอบประสิทธิภาพในการประเมินความมั่นคงปลอดภัยและการตรวจหาช่องโหว่ของเว็บไซต์ Moodle ได้อย่างมีคุณภาพ โดยสามารถแบ่งเป็นหัวข้อต่าง ๆ ได้ดังนี้

2.1 ระบบการจัดการการเรียนรู้ (LMS) และ Moodle
2.2 ความมั่นคงปลอดภัยของเว็บไซต์ Moodle
2.3 ช่องโหว่ที่มีอยู่ในเว็บไซต์ Moodle ของทั้ง 2 เวอร์ชัน
2.4 การโจมตีทางไซเบอร์ที่พบบ่อยในระบบ LMS
2.5 เครื่องมือที่ใช้ในการทดสอบความมั่นคงปลอดภัย
2.6 แนวทางการป้องกันช่องโหว่ด้านความมั่นคงปลอดภัยในเว็บแอปพลิเคชัน
2.7 การใช้ Docker และ Ubuntu ในการทดสอบ Moodle
2.8 งานวิจัยที่เกี่ยวข้อง

2.1 ระบบการจัดการการเรียนรู้ (LMS) และ Moodle

ระบบการจัดการการเรียนรู้ (Learning Management System LMS) เป็นซอฟต์แวร์ที่ช่วยบริหารจัดการการเรียนการสอนผ่านระบบออนไลน์ LMS มีความสำคัญอย่างมากในยุคดิจิทัล เนื่องจากช่วยให้สามารถจัดหลักสูตร อัปโหลดเนื้อหา จัดการข้อสอบ และติดตามผลการเรียนของผู้เรียนได้อย่างเป็นระบบทำให้การเรียนการสอนมีความสะดวก และมีประสิทธิภาพมากขึ้น โดยเฉพาะในช่วง ที่การเรียนการสอนออนไลน์มีบทบาทสำคัญมากขึ้นในวิทยาลัยการศึกษา และภาคธุรกิจ

2.1.1 คุณสมบัติหลักของ LMS ได้แก่

2.1.1.1 การจัดการหลักสูตร (Course Management) รองรับการสร้างและจัดการเนื้อหาการเรียนรู้ เช่น เอกสาร วิดีโอ และแบบฝึกหัดออนไลน์

2.1.1.2 การติดตามและประเมินผล (Tracking & Assessment) ระบบสามารถติดตามความก้าวหน้าของผู้เรียน รวมถึงจัดการข้อสอบออนไลน์ได้

2.1.1.3 การสื่อสารและปฏิสัมพันธ์ (Communication & Collaboration) รองรับการสื่อสารระหว่างผู้สอนและผู้เรียนผ่านฟอรัม แชท หรืออีเมล

2.1.1.4 การจัดการผู้ใช้ (User Management) กำหนดบทบาทและสิทธิ์ของนักเรียน อาจารย์ และผู้ดูแลระบบได้อย่างยืดหยุ่น

LMS ถูกนำมาใช้ในวิทยาลัยการศึกษาทั่วโลก รวมถึงองค์กรภาคธุรกิจที่ต้องการฝึกอบรมพนักงาน และเป็นรากฐานสำคัญของการศึกษาแบบ e-Learning [1]

2.1.2 การทำงานของ Moodle และความนิยมในการใช้งาน

Moodle (Modular Object-Oriented Dynamic Learning Environment) เป็นระบบ LMS แบบโอเพนซอร์สที่ได้รับความนิยมมาก เนื่องจากสามารถปรับแต่งและขยายฟังก์ชันการทำงานได้ตามความต้องการของผู้ใช้ Moodle ถูกออกแบบมาให้รองรับทั้งการเรียนรู้แบบห้องเรียนออนไลน์และการฝึกอบรมในองค์กร โดยเป็นหนึ่งใน LMS ที่ถูกใช้งานมากที่สุดในมหาวิทยาลัยทั่วโลก [1]

2.1.2.1 คุณสมบัติหลักของ Moodle

2.1.2.1.1 โครงสร้างแบบโมดูลาร์ รองรับการเพิ่มปลั๊กอิน เช่น ระบบข้อสอบ (Quiz) ระบบส่งงาน (Assignment) และ Gamification

2.1.2.1.2 การจัดการหลักสูตร รองรับการเรียนแบบ Self-paced และ Instructor-led

2.1.2.1.3 การโต้ตอบและการมีส่วนร่วมของผู้เรียน รองรับการสื่อสารผ่านฟอรัม การส่งข้อความ และวิดีโอคอนเฟอเรนซ์

2.1.2.1.4 มาตรฐาน SCORM และ LTI รองรับการเชื่อมต่อกับแพลตฟอร์มการเรียนรู้อื่น ๆ

2.1.2.1.5 การจัดการผู้ใช้ที่ยืดหยุ่น รองรับการกำหนดบทบาท เช่น ผู้สอน ผู้เรียน และผู้ดูแลระบบ

2.1.2.2 ข้อดีของ Moodle

2.1.2.2.1 เป็นซอฟต์แวร์โอเพนซอร์สที่ไม่มีค่าใช้จ่าย สามารถดาวน์โหลดและปรับแต่งได้ฟรี

2.1.2.2.2 รองรับหลายภาษา ใช้งานได้ในหลายประเทศ

2.1.2.2.3 รองรับการใช้งานบนอุปกรณ์พกพา

2.1.2.2.4 มีชุมชนผู้ใช้ขนาดใหญ่ที่ช่วยกันพัฒนาและสนับสนุน

2.2 ความมั่นคงปลอดภัยของเว็บไซต์ Moodle

Moodle เป็นระบบจัดการการเรียนรู้แบบโอเพนซอร์สที่ได้รับความนิยมทั่วโลก เนื่องจากเป็นแพลตฟอร์มที่เปิดให้สามารถปรับแต่ง และพัฒนาเพิ่มเติมได้ ซึ่งทำให้เกิดข้อได้เปรียบในการใช้งาน อย่างไรก็ตาม การเปิดกว้างดังกล่าวอาจนำไปสู่ความเสี่ยงด้านความมั่นคงปลอดภัยหากไม่มีการกำหนดมาตรการป้องกันที่เหมาะสม หลักการสำคัญของความมั่นคงปลอดภัยใน Moodle ประกอบด้วย

2.2.1 Authentication & Authorization ระบบต้องมีการยืนยันตัวตนที่เข้มงวด และกำหนดสิทธิ์การเข้าถึงข้อมูลให้เหมาะสมกับบทบาทของผู้ใช้ เช่น นักเรียน อาจารย์ และผู้ดูแลระบบ [7]

2.2.2 Data Encryption ควรมีการใช้ SSL/TLS เพื่อเข้ารหัสข้อมูลที่รับส่งระหว่างเซิร์ฟเวอร์กับไคลเอนต์ เพื่อลดความเสี่ยงจากการถูกดักจับข้อมูล [9]

2.2.3 Input Validation & Filtering ควรมีการตรวจสอบข้อมูลที่ป้อนเข้ามาในระบบเพื่อป้องกัน SQL Injection และ Cross-Site Scripting (XSS) [5]

2.2.4 Security Patching & Updates ควรมีการอัปเดตซอฟต์แวร์และแพตช์ช่องโหว่ของระบบ Moodle อย่างสม่ำเสมอเพื่อป้องกันการโจมตีจากช่องโหว่ที่เป็นที่รู้จักแล้ว [14]

2.2.5 แนวทางการตั้งค่าความมั่นคงปลอดภัยของ Moodle มีฟีเจอร์ด้านความมั่นคงปลอดภัยที่สามารถตั้งค่าให้เหมาะสมกับการใช้งาน โดยแนวทางการตั้งค่าที่แนะนำ ได้แก่

2.2.5.1 การตั้งค่าการเข้าถึงและการยืนยันตัวตน

2.2.5.1.1 เปิดใช้งาน Multi-Factor Authentication (MFA) เพื่อเพิ่มระดับความมั่นคงปลอดภัยในการล็อกอิน

2.2.5.1.2 ใช้ Strong Password Policy บังคับให้ผู้ใช้ตั้งรหัสผ่านที่ซับซ้อน

2.2.5.1.3 จำกัดจำนวนครั้งในการลองล็อกอินผิดพลาดเพื่อลดความเสี่ยงจากการโจมตีแบบ Brute Force [6]

2.2.5.2 การปกป้องข้อมูลและการเข้ารหัส

2.2.5.2.1 เปิดใช้งาน HTTPS (SSL/TLS) เพื่อเข้ารหัสข้อมูลที่รับส่งระหว่างผู้ใช้กับเซิร์ฟเวอร์

2.2.5.2.2 เปิดใช้งาน Database Encryption สำหรับข้อมูลที่สำคัญ เช่น รหัสผ่านและข้อมูลส่วนบุคคลของนักเรียน [9]

2.3 การโจมตีทางไซเบอร์ที่พบบ่อยในระบบ LMS

2.3.1 การโจมตีแบบ Brute Force บนหน้า Login การโจมตีแบบ Brute Force เป็นการสุ่มรหัสผ่านเพื่อพยายามล็อกอินเข้าสู่ระบบ Moodle โดยการทดลองใส่รหัสผ่านจำนวนมากจนกว่าจะสามารถเข้าสู่ระบบได้สำเร็จ วิธีการป้องกัน ได้แก่

2.3.1.1 เปิดใช้งาน Account Lockout หลังจากมีความพยายามเข้าสู่ระบบผิดพลาดหลายครั้ง

2.3.1.2 ใช้ CAPTCHA เพื่อป้องกันบอทจากการสุ่มรหัสผ่าน

2.3.1.3 ใช้ Multi-Factor Authentication เพื่อเพิ่มการยืนยันตัวตนอีกชั้นหนึ่ง [6]

2.3.2 การใช้ SQL Injection เพื่อเข้าถึงฐานข้อมูล SQL Injection เป็นช่องโหว่ที่พบได้บ่อยในเว็บแอปพลิเคชันที่มีการรับค่าจากผู้ใช้โดยไม่มีการตรวจสอบที่เพียงพอ ผู้โจมตีสามารถส่งคำสั่ง SQL อันตรายเพื่อเข้าถึง แก้ไข หรือทำลายข้อมูลในฐานข้อมูลของ Moodle วิธีการป้องกัน ได้แก่

2.3.2.1 ใช้ Parameterized Queries แทนการใช้ SQL Query โดยตรง

2.3.2.2 เปิดใช้งาน Web Application Firewall (WAF) เพื่อป้องกันคำสั่ง SQL อันตราย [15]

2.3.2.3 ใช้เครื่องมือ SQLMap เพื่อตรวจหาช่องโหว่ SQL Injection และแก้ไขจุดอ่อนในโค้ด [16]

2.3.3 การโจมตีแบบ Cross-Site Scripting (XSS) และผลกระทบต่อ Moodle เป็นการโจมตีที่อาศัยช่องโหว่ในการรับค่าข้อมูลจากผู้ใช้ และแสดงผลกลับโดยไม่มีการกรองข้อมูล ทำให้ผู้โจมตีสามารถฝังโค้ด JavaScript ที่เป็นอันตรายลงไปในหน้าเว็บของ Moodle ได้ วิธีการป้องกัน ได้แก่

2.3.3.1 ใช้ Input Validation และ Output Encoding เพื่อตรวจสอบและป้องกันการแสดงผลของโค้ดที่เป็นอันตราย

2.3.3.2 ใช้ Content Security Policy (CSP) เพื่อลดความสามารถของโค้ด JavaScript ที่ไม่พึงประสงค์ [12]

2.3.4 ช่องโหว่ด้านการอัปโหลดไฟล์และ Remote Code Execution Moodle อนุญาตให้ผู้ใช้สามารถอัปโหลดไฟล์ เช่น ไฟล์เอกสารและสื่อการเรียนรู้ ซึ่งอาจถูกใช้เป็นช่องทางในการโจมตีได้ โดยผู้โจมตีอาจอัปโหลดไฟล์ที่มีโค้ดอันตราย (เช่น .php, .exe) และดำเนินการรันโค้ดบนเซิร์ฟเวอร์ Moodle วิธีการป้องกัน ได้แก่

2.3.4.1 กำหนด Whitelist File Extensions เพื่อให้ Moodle ยอมรับเฉพาะไฟล์ที่ปลอดภัย

2.3.4.2 เปิดใช้งาน Antivirus Scanner เพื่อตรวจสอบไฟล์ก่อนการอัปโหลด

2.3.4.3 ใช้ Chroot Jail และ Least Privilege Access เพื่อจำกัดสิทธิ์ของผู้ใช้ในระบบ [14]

2.4 เครื่องมือที่ใช้ในการทดสอบความมั่นคงปลอดภัย

ในการทดสอบความมั่นคงปลอดภัยของระบบ Moodle จำเป็นต้องใช้เครื่องมือหลายประเภทเพื่อสแกนหาช่องโหว่และประเมินความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน เครื่องมือที่ใช้ในการทดสอบช่องโหว่และความมั่นคงปลอดภัยของระบบ Moodle มีดังนี้

2.4.1 OWASP ZAP (Zed Attack Proxy) เป็นเครื่องมือโอเพนซอร์สที่ใช้ในการสแกนและวิเคราะห์ช่องโหว่ของเว็บแอปพลิเคชัน โดยเฉพาะช่องโหว่ที่พบบ่อย เช่น SQL Injection, Cross-Site Scripting (XSS) และ Remote Code Execution (RCE) โดยเครื่องมือสามารถทำ Active Scan และ Passive Scan เพื่อวิเคราะห์ช่องโหว่ในแอปพลิเคชัน Moodle โดยไม่ต้องเข้าถึงโค้ดต้นทาง [4]

2.4.2 OpenVAS (Open Vulnerability Assessment System) OpenVAS เป็นเครื่องมือโอเพนซอร์สสำหรับการประเมินช่องโหว่และการจัดการความมั่นคงปลอดภัยของระบบเครือข่าย โดยเน้นการสแกนหาช่องโหว่ที่อาจถูกโจมตีในระบบเซิร์ฟเวอร์ ฐานข้อมูล และแอปพลิเคชันต่าง ๆ OpenVAS สามารถตรวจจับช่องโหว่ที่หลากหลาย รวมถึงการตั้งค่าระบบที่ไม่ปลอดภัย และมีฐานข้อมูลช่องโหว่ที่อัปเดตอยู่เสมอ ซึ่งช่วยให้การทดสอบความมั่นคงปลอดภัยของ Moodle มีประสิทธิภาพมากขึ้น [5]

2.4.3 Nessus เป็นเครื่องมือสแกนช่องโหว่เชิงพาณิชย์ที่ได้รับความนิยมอย่างแพร่หลายในการทดสอบความมั่นคงปลอดภัยของระบบเครือข่าย และแอปพลิเคชัน โดย Nessus สามารถตรวจจับช่องโหว่ในระบบได้หลากหลาย เช่น การโจมตีแบบ SQL Injection การตั้งค่าระบบที่ไม่

ปลอดภัย และการจัดการสิทธิ์ที่ไม่เหมาะสม นอกจากนี้ Nessus ยังมีระบบรายงานผลที่ช่วยให้สามารถวิเคราะห์ และจัดการช่องโหว่ได้อย่างมีประสิทธิภาพ [9]

2.5 แนวทางการป้องกันช่องโหว่ด้านความมั่นคงปลอดภัยในเว็บแอปพลิเคชัน

การรักษาความมั่นคงปลอดภัยของ Moodle และเว็บแอปพลิเคชันที่เกี่ยวข้องมีความสำคัญอย่างยิ่ง เนื่องจากเป็นระบบที่จัดเก็บข้อมูลที่สำคัญ เช่น ข้อมูลของนักศึกษา อาจารย์ และเนื้อหาการเรียนรู้ต่าง ๆ ซึ่งหากระบบมีช่องโหว่ อาจทำให้เกิดการรั่วไหลของข้อมูล หรือถูกโจมตีโดยแฮ็กเกอร์ได้ แนวทางการป้องกันช่องโหว่ที่สำคัญประกอบด้วย

2.5.1 การตั้งค่า Content Security Policy (CSP) เป็นกลไกความมั่นคงปลอดภัยที่ช่วยลดความเสี่ยงจากการโจมตีประเภท Cross-Site Scripting (XSS) และ Data Injection Attacks โดย CSP จะช่วยป้องกัน ไม่ให้เว็บเบราว์เซอร์โหลดเนื้อหาหรือสคริปต์จากแหล่งที่ไม่น่าเชื่อถือ ซึ่งช่วยป้องกันการแทรกโค้ดอันตรายลงในหน้าเว็บของ Moodle

2.5.2 การใช้งาน SSL/TLS (Secure Sockets Layer / Transport Layer Security) เป็นโปรโตคอลที่ใช้เข้ารหัสข้อมูลที่รับ-ส่งระหว่างผู้ใช้และเซิร์ฟเวอร์ ซึ่งช่วยป้องกันการดักจับข้อมูลระหว่างทาง (Man-in-the-Middle Attack, MITM) การใช้ HTTPS ที่เปิดใช้งาน SSL/TLS เป็นสิ่งที่จำเป็นสำหรับ Moodle เพื่อเพิ่มความมั่นคงปลอดภัยให้กับระบบ

2.5.3 การใช้ Multi-Factor Authentication (MFA) เป็นกลไกที่ช่วยเพิ่มระดับความมั่นคงปลอดภัยในการล็อกอินเข้าสู่ระบบ Moodle โดยบังคับให้ผู้ใช้ต้องยืนยันตัวตนมากกว่าหนึ่งปัจจัย ซึ่งช่วยป้องกันการเข้าถึงบัญชีโดยไม่ได้รับอนุญาต แม้ว่ารหัสผ่านจะรั่วไหล

2.6 การใช้ Docker และ Ubuntu ในการทดสอบ Moodle

Docker เป็นแพลตฟอร์มที่ช่วยให้สามารถสร้าง และจัดการสภาพแวดล้อมแบบคอนเทนเนอร์ (Containerized Environment) ได้อย่างสะดวก และยืดหยุ่นทำให้สามารถติดตั้ง และทดสอบ Moodle เวอร์ชันต่างๆ ได้โดยไม่ต้องติดตั้งซอฟต์แวร์ทั้งหมดลงบนเครื่องเซิร์ฟเวอร์โดยตรง ช่วยลดปัญหาความขัดแย้งของไลบรารีหรือเวอร์ชันของซอฟต์แวร์ที่ใช้งานร่วมกัน นอกจากนี้ ยังประหยัดทรัพยากรใช้หน่วยความจำและพื้นที่เก็บข้อมูลน้อยกว่าการติดตั้งแบบปกติ

ในการทดสอบครั้งนี้ มีการติดตั้ง Moodle 3.7.1 และ Moodle 4.5.1 ผ่าน Docker บนระบบปฏิบัติการ Ubuntu 24.04.1 LTS ที่ทำงานบน Proxmox โดยมีการตั้งค่า Docker Compose เพื่อจำลองเซิร์ฟเวอร์ที่มีความเสถียร และสามารถบริหารจัดการได้ง่าย Ubuntu 24.04.1 LTS เป็นระบบปฏิบัติการที่ได้รับความนิยมสำหรับการใช้งานในเซิร์ฟเวอร์ โดยรองรับ Docker ได้ดี และมีข้อดีหลายประการที่ช่วยในการทดสอบ Moodle เช่น รองรับการใช้งานระยะยาว (Long Term Support LTS) มีการอัปเดตความมั่นคงปลอดภัย และการสนับสนุนอย่างต่อเนื่อง รองรับ Docker และ Virtualization สามารถใช้งาน Docker, LXD และ KVM ได้ดี มีเสถียรภาพสูงรองรับการติดตั้งบน

Proxmox ทำให้สามารถจำลองเซิร์ฟเวอร์เสมือนได้ มีเครื่องมือบริหารจัดการครบครันสามารถติดตั้งเครื่องมือสำหรับตรวจสอบระบบ และบำรุงรักษาได้ง่าย เป็นต้น

2.7 งานวิจัยที่เกี่ยวข้อง

จากการศึกษาโดย Samgir และคณะ [4] และ Kunda และคณะ [9] พบว่าเครื่องมือ Metasploit และ OWASP ZAP มีบทบาทสำคัญในการตรวจจับและวิเคราะห์ช่องโหว่ที่พบบ่อย เช่น SQL Injection, Cross-Site Scripting (XSS) และ Remote Code Execution (RCE) โดย Metasploit ทำหน้าที่เป็นสภาพแวดล้อมจำลองการโจมตี ในขณะที่ OWASP ZAP ใช้สำหรับการสแกนช่องโหว่ในเว็บแอปพลิเคชัน การผสานการทำงานของเครื่องมือทั้งสองช่วยลดเวลาในการทดสอบและเพิ่มประสิทธิภาพในการตรวจจับช่องโหว่ได้อย่างมีประสิทธิภาพ

งานวิจัยโดย Siva Lakshmi และคณะ [5], Qingyang และคณะ [12] และ Lv และคณะ [13] ได้เสนอแนวทางการป้องกัน SQL Injection และ XSS ผ่านการกรองข้อมูล การเข้ารหัส และการใช้เทคนิค Fuzzing Test และ Adaptive Random Testing (ART) เพื่อวิเคราะห์การตอบสนองของระบบเว็บแอปพลิเคชัน โดยการใช้เทคนิค ART ช่วยเพิ่มประสิทธิภาพในการตรวจจับจุดเสี่ยงจากการโจมตีแบบ XSS ได้ดีกว่าการใช้ Fuzzing แบบเดิม

จากการศึกษาของ AlMufairej และคณะ [6], Ameer และคณะ [8] และ Patel และคณะ [7] พบว่าการจัดการความเสี่ยงทางไซเบอร์ในระบบ E-Learning และ IoT สามารถทำได้ผ่านการใช้โมเดลการควบคุมการเข้าถึง เช่น RBAC (Role-Based Access Control) และ ABAC (Attribute-Based Access Control) โดยงานวิจัยเหล่านี้เสนอแนวทางการใช้ไฟร์วอลล์ การเข้ารหัสข้อมูล และการอบรมผู้ใช้งานเพื่อจัดการความเสี่ยง รวมถึงการผสมผสาน ABAC และ RBAC เพื่อเพิ่มความยืดหยุ่นในการกำหนดสิทธิ์ของผู้ใช้

งานวิจัยโดย Tyagi และคณะ [10], Shahid และคณะ [11] และ Subbulakshmi และคณะ [15] ได้ทำการประเมินเครื่องมือวิเคราะห์ช่องโหว่ เช่น OWASP WAP, RIPS, SQLMAP และ SAT ซึ่งใช้ตรวจจับช่องโหว่เว็บแอปพลิเคชัน ผลการศึกษาแสดงให้เห็นว่า SAT ที่ใช้เทคโนโลยี Machine Learning สามารถตรวจจับช่องโหว่ได้แม่นยำและครอบคลุมมากกว่าเครื่องมือแบบดั้งเดิม ในขณะที่ SQLMAP และ OWASP WAP มีประสิทธิภาพดีในการตรวจจับช่องโหว่ที่เกี่ยวข้องกับ SQL Injection และ XSS

จากการศึกษาของ Zabala และคณะ [14] และ Min และคณะ [17] ได้เสนอโมเดลความมั่นคงปลอดภัยแบบหลายชั้น (Multi-layered Security Model) สำหรับระบบ Moodle เช่น ชั้นของแอปพลิเคชัน (Application Layer) ฐานข้อมูล (Database Layer) และไฟล์ (File Layer) โดยการใช้โมเดลนี้ช่วยลดความเสี่ยงจาก SQL Injection และ XSS รวมถึงการป้องกันข้อมูลสูญหายผ่านการสำรองข้อมูลและการตรวจสอบความถูกต้องของไฟล์

Abdullah และคณะ [16] ได้ทำการทดสอบประสิทธิภาพของเครื่องมือ SQLMAP ในการตรวจจับและโจมตีช่องโหว่แบบ SQL Injection โดยใช้เว็บแอปพลิเคชันที่มีช่องโหว่โดยเจตนาอย่าง

Damn Vulnerable Web Application (DVWA) ผลการศึกษาชี้ให้เห็นว่า SQLMAP สามารถระบุฐานข้อมูล backend และดึงข้อมูลสำคัญออกมาได้อย่างมีประสิทธิภาพ ผ่านการใช้คำสั่ง SQLMAP เพื่อทำการเจาะระบบฐานข้อมูล การทดลองนี้ยังเน้นถึงความสำคัญของการใช้มาตรการป้องกันที่เหมาะสม เช่น การตรวจสอบข้อมูลนำเข้า (Input Validation) และการใช้คำสั่งแบบ Parameterized Queries เพื่อป้องกันการฉีด SQL ที่เป็นอันตราย

จากการศึกษางานวิจัยที่เกี่ยวข้อง ผู้วิจัยได้นำแนวทางการใช้ OWASP ZAP และ Nessus มาช่วยในการตรวจจับช่องโหว่ของ Moodle โดยเน้นการวิเคราะห์ SQL Injection (SQLi), Cross-Site Scripting (XSS) และ Remote Code Execution (RCE) รวมถึงการใช้แนวทาง การควบคุมการเข้าถึงระบบ ผ่าน RBAC (Role-Based Access Control) เพื่อเสริมความมั่นคงปลอดภัยในการบริหารสิทธิ์ผู้ใช้งาน นอกจากนี้ ยังได้นำเทคนิค Fuzzing Test และ Adaptive Random Testing (ART) มาใช้ในการวิเคราะห์จุดอ่อนของระบบ รวมถึงการตั้งค่า Security Headers, Content Security Policy (CSP) และ SSL/TLS เพื่อป้องกันการโจมตีทางไซเบอร์ อีกทั้งยังอ้างอิง โมเดลความมั่นคงปลอดภัยแบบหลายชั้น (Multi-layered Security Model) สำหรับระบบ Moodle เพื่อลดความเสี่ยงจากช่องโหว่ด้านฐานข้อมูลและไฟล์ ซึ่งช่วยให้สามารถปรับปรุงมาตรการป้องกันของระบบ Moodle ได้อย่างมีประสิทธิภาพมากขึ้น

บทที่ 3
วิธีการดำเนินงานวิจัย

การดำเนินการวิจัยนี้มุ่งเน้นไปที่การประเมินความมั่นคงปลอดภัยของเว็บไซต์ Moodle ผ่านกระบวนการทดสอบช่องโหว่ที่เป็นระบบ โดยใช้เครื่องมือโอเพนซอร์ส และสภาพแวดล้อมจำลองเพื่อทดสอบช่องโหว่ที่อาจเกิดขึ้น Moodle 3.7.1 และ Moodle 4.5.1 ได้รับการติดตั้งในระบบ Docker บน Ubuntu 24.04.1 LTS ที่ทำงานบน Proxmox โดยมีการออกแบบ และดำเนินการทดสอบอย่างเป็นขั้นตอนในการดำเนินการวิจัยได้ดังต่อไปนี้

3.1 ศึกษาและวิเคราะห์ปัญหาที่เกี่ยวข้องกับงานวิจัย

3.2 การเตรียมเครื่องมือและสภาพแวดล้อมการทดสอบ

3.3 การเตรียมและจัดการข้อมูล

3.4 การทดสอบช่องโหว่และการประเมินความเสี่ยง

3.1 ศึกษาและวิเคราะห์ปัญหาที่เกี่ยวข้องกับงานวิจัย

การศึกษานี้มุ่งเน้นไปที่การตรวจสอบช่องโหว่ที่อาจส่งผลกระทบต่อระบบ Moodle โดยใช้แนวทางการวิเคราะห์สถาปัตยกรรมและการทำงานของ Moodle 3.7.1 และ Moodle 4.5.1 เพื่อทำความเข้าใจถึงปัญหาด้านความมั่นคงปลอดภัยที่อาจเกิดขึ้นรวมถึงศึกษาวิธีการป้องกัน และลดความเสี่ยงในการถูกโจมตี โดยแบ่งดังนี้

3.1.1 ศึกษาระบบการจัดการเรียนรู้ (LMS) และสถาปัตยกรรมของ Moodle

ระบบการจัดการเรียนรู้ (Learning Management System - LMS) เป็นแพลตฟอร์มที่ช่วยให้การบริหารจัดการหลักสูตรออนไลน์มีความสะดวกยิ่งขึ้น Moodle เป็นหนึ่งใน LMS ที่ได้รับความนิยม เนื่องจากเป็นโอเพนซอร์ส (Open Source) และสามารถปรับแต่งได้ตามความต้องการของวิทยาลัยการศึกษา และองค์กรต่าง ๆ สถาปัตยกรรมของ Moodle ประกอบด้วยองค์ประกอบหลักได้แก่

3.1.1.1 Core System เป็นศูนย์กลางของระบบที่จัดการหลักสูตร บทเรียน การสอบและการส่งงาน

3.1.1.2 Database Layer รองรับฐานข้อมูลหลากหลาย เช่น MySQL, PostgreSQL และ MariaDB

3.1.1.3 Web Server ทำงานร่วมกับ Apache หรือ Nginx เพื่อให้บริการเนื้อหาผ่าน HTTP/HTTPS

3.1.1.4 Authentication & Authorization รองรับการยืนยันตัวตน เช่น LDAP, OAuth2 และ Single Sign-On (SSO)

3.1.2 วิเคราะห์ช่องโหว่ที่มีการรายงานใน Moodle 3.7.1 และ Moodle 4.5.1

เนื่องจาก Moodle เป็น โอเพนซอร์สซอฟต์แวร์ จึงมีการปรับปรุงความมั่นคงปลอดภัยอย่างต่อเนื่อง โดยมีการอัปเดตแพตช์และการแก้ไขช่องโหว่ในแต่ละเวอร์ชัน การศึกษานี้ได้รวบรวม และวิเคราะห์ช่องโหว่ที่พบใน Moodle 3.7.1 และ Moodle 4.5.1 เพื่อเปรียบเทียบว่ามีการปรับปรุงด้านความมั่นคงปลอดภัยอย่างไรบ้าง ตัวอย่างช่องโหว่ที่พบใน Moodle 3.7.1 และ Moodle 4.5.1 ได้แก่

3.1.2.1 ช่องโหว่ด้านการจัดการอินพุต (Input Validation Issues)

3.1.2.2 ช่องโหว่ที่เกี่ยวข้องกับการตรวจสอบสิทธิ์ของผู้ใช้ (Access Control Issues)

3.1.2.3 ช่องโหว่ที่เกิดจากการตั้งค่าระบบที่ไม่ปลอดภัย (Misconfiguration Issues)

3.1.2.4 ช่องโหว่ที่เกี่ยวข้องกับ API และ Web Services

3.1.3 ศึกษาแนวทางการทดสอบช่องโหว่ที่เกี่ยวข้อง

เพื่อให้การวิเคราะห์ความมั่นคงปลอดภัยของ Moodle มีประสิทธิภาพสูงสุด การศึกษานี้มุ่งเน้นไปที่การตรวจสอบช่องโหว่ที่มักพบในระบบ LMS โดยใช้เครื่องมือที่เป็นมาตรฐานสำหรับการทดสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน ช่องโหว่ที่ศึกษาได้แก่

3.1.3.1 SQL Injection เป็นช่องโหว่ที่เกิดขึ้นจากการที่เว็บแอปพลิเคชันไม่ได้ทำการกรองอินพุตจากผู้ใช้ ทำให้สามารถรันคำสั่ง SQL เพื่อเข้าถึงหรือแก้ไขฐานข้อมูลโดยไม่ได้รับอนุญาต

3.1.3.2 Cross-Site Scripting (XSS) เป็นการโจมตีที่แฮ็กเกอร์สามารถฝังโค้ด JavaScript อันตรายลงในหน้าเว็บของ Moodle ซึ่งอาจถูกใช้เพื่อขโมยข้อมูลการเข้าสู่ระบบของผู้ใช้

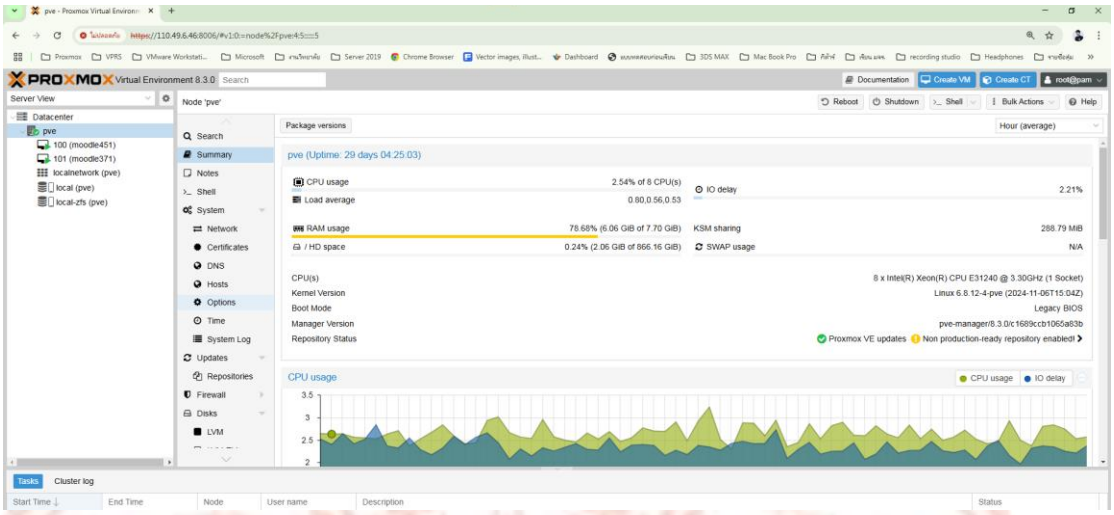
3.1.3.3 Remote Code Execution (RCE) ช่องโหว่นี้ช่วยให้ผู้โจมตีสามารถรันคำสั่งบนเซิร์ฟเวอร์ของ Moodle ได้ ซึ่งอาจนำไปสู่การเข้าควบคุมระบบ

3.1.3.4 ช่องโหว่ด้านการอัปโหลดไฟล์ที่ไม่ปลอดภัย Moodle อนุญาตให้ผู้ใช้สามารถอัปโหลดไฟล์ได้ ซึ่งหากไม่มีการตรวจสอบที่ดีพอ อาจทำให้แฮ็กเกอร์สามารถอัปโหลดไฟล์ที่มีโค้ดอันตราย และรันคำสั่งบนเซิร์ฟเวอร์ได้

3.2 การเตรียมเครื่องมือและสภาพแวดล้อมการทดสอบ

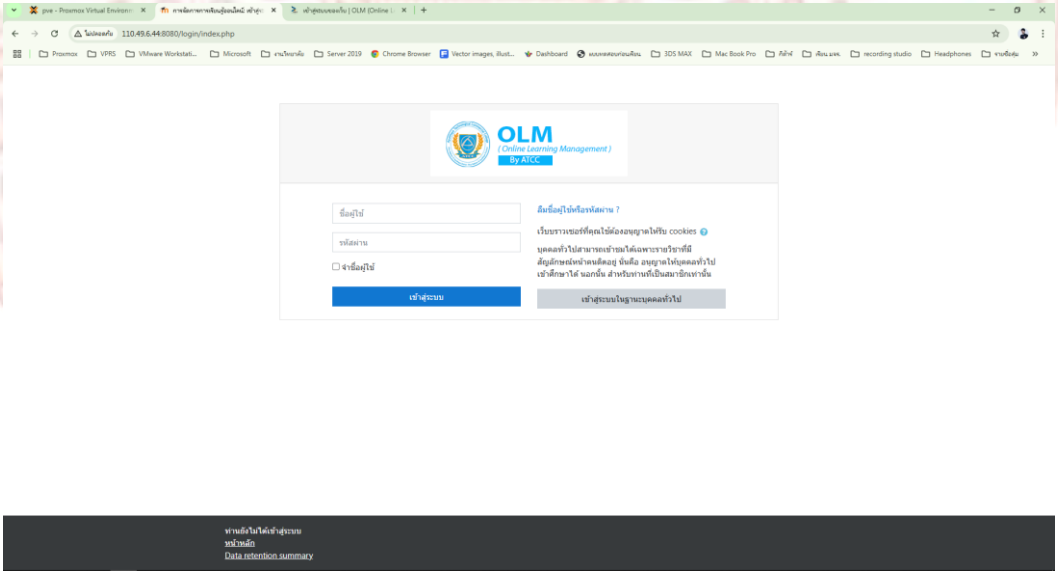
3.2.1 การจำลองระบบปฏิบัติการ Linux

ใช้ Proxmox ในการจำลองระบบปฏิบัติการ Linux บน HP ProLiant DL120 G7 เป็นเซิร์ฟเวอร์ดังกล่าวถูกใช้เป็นโครงสร้างหลักในการจำลองระบบเสมือนจริงสำหรับการติดตั้ง และทดสอบระบบ Moodle รายละเอียดของเซิร์ฟเวอร์ที่ใช้แสดงใน ภาพที่ 3-1

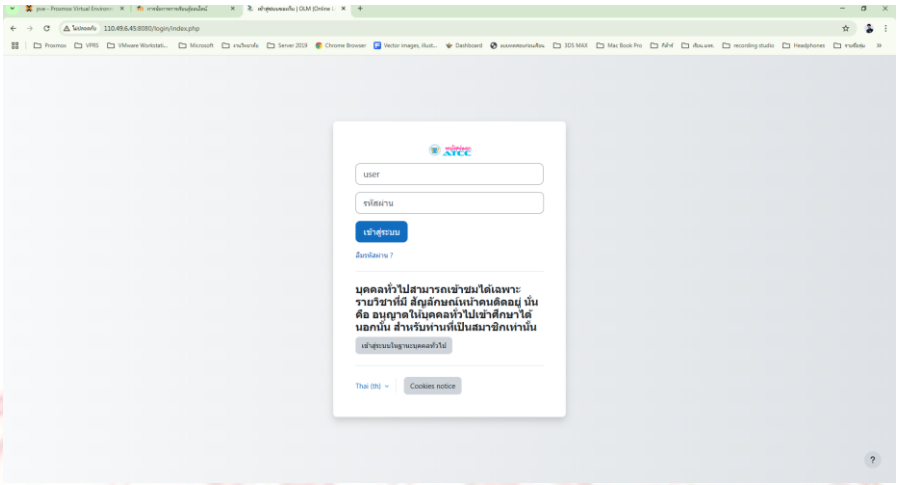


ภาพที่ 3-1 เซิร์ฟเวอร์หลักในการจำลองระบบ Proxmox

เซิร์ฟเวอร์หลักในการจำลองระบบใช้ Intel Xeon E31240 (Quad Core) ที่มี Clock Speed 2.868 GHz ซึ่งหลังจากการติดตั้งระบบปฏิบัติการ ได้มีการดำเนินการติดตั้ง Moodle 3.7.1 และ Moodle 4.5.1 เพื่อใช้เป็นแพลตฟอร์มในการทดสอบช่องโหว่ และประเมินความมั่นคงปลอดภัยของระบบ Moodle ซึ่งหน้าตาของเว็บแต่ละเวอร์ชันดัง ภาพที่ 3-2 และ ภาพที่ 3-3



ภาพที่ 3-2 ระบบ Moodle 3.7.1 ที่ใช้ทดสอบ



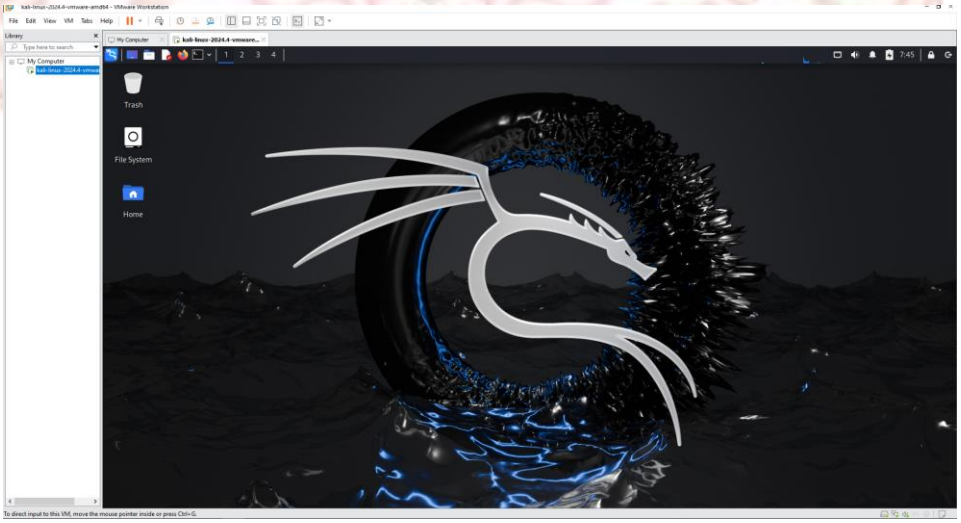
ภาพที่ 3-3 ระบบ Moodle 4.5.1 ที่ใช้ทดสอบ

3.2.2 การจำลองระบบปฏิบัติการ Linux สำหรับการตรวจสอบช่องโหว่ด้านความมั่นคงปลอดภัย

เพื่อสนับสนุนการทดสอบความมั่นคงปลอดภัยของระบบ Moodle 3.7.1 และ Moodle 4.5.1 ได้มีการใช้ VirtualBox หรือ VMware ในการจำลองระบบปฏิบัติการ Linux บน DELL Precision 5530 ซึ่งถูกใช้เป็นเครื่องทดสอบหลักสำหรับการวิเคราะห์และประเมินช่องโหว่ของระบบ Moodle

ในการทดสอบเจาะระบบ Kali Linux ถูกติดตั้งภายในสภาพแวดล้อมเสมือนเพื่อใช้เป็นเครื่องมือหลักในการตรวจสอบและจำลองการโจมตีดังแสดงในภาพที่ 3-4 โดย Kali Linux เป็นระบบปฏิบัติการที่ออกแบบมาโดยเฉพาะสำหรับการทดสอบความมั่นคงปลอดภัย ซึ่งมาพร้อมเครื่องมือสำหรับ การสแกนช่องโหว่ การโจมตีแบบเจาะระบบ และการประเมินความเสี่ยงอย่างครอบคลุม

การจำลองระบบช่วยให้สามารถทดสอบการโจมตีทางไซเบอร์ในสภาพแวดล้อมที่ปลอดภัย ระบุจุดอ่อนของ Moodle ก่อนนำไปใช้งานจริง และพัฒนาแนวทางป้องกันที่มีประสิทธิภาพมากขึ้น



ภาพที่ 3-4 ติดตั้ง Kali Linux สำหรับทดสอบเจาะระบบ Moodle

3.3 การเตรียมและจัดการข้อมูล

รวบรวมข้อมูลจาก System Log ฟอร์มลงทะเบียน การล็อกอิน การอัปโหลดไฟล์ การเรียกใช้งาน URL โดยข้อมูลที่ได้ถูกนำมาวิเคราะห์เพื่อประเมินความมั่นคงปลอดภัยของ Moodle และตรวจสอบช่องโหว่ด้าน Input Validation, Authentication และ Access Control

3.4 การทดสอบช่องโหว่และการประเมินความเสี่ยง

3.4.1 เครื่องมือที่ใช้ในการสแกนช่องโหว่

เพื่อประเมินความมั่นคงปลอดภัยของระบบ Moodle การทดสอบช่องโหว่ดำเนินการโดยใช้เครื่องมือสแกนช่องโหว่ 3 ประเภท ซึ่งครอบคลุมทั้ง เว็บแอปพลิเคชัน ระบบเซิร์ฟเวอร์ และเครือข่าย ดังนี้

3.4.1.1 OWASP ZAP ตรวจสอบช่องโหว่ใน เว็บแอปพลิเคชัน เช่น SQL Injection, Cross-Site Scripting (XSS) และการตั้งค่าความมั่นคงปลอดภัยของเว็บ

3.4.1.2 OpenVAS วิเคราะห์และประเมินช่องโหว่ของ ระบบเซิร์ฟเวอร์ และการกำหนดค่าความมั่นคงปลอดภัย

3.4.1.3 Nessus ตรวจสอบช่องโหว่ด้าน เครือข่ายและระบบปฏิบัติการ วิเคราะห์ความเสี่ยงจากช่องโหว่ของซอฟต์แวร์และโปรโตคอลการสื่อสาร

3.4.2 การจำแนกประเภทของช่องโหว่

3.4.2.1 Critical ช่องโหว่ที่มีความเสี่ยงร้ายแรงที่สุด อาจส่งผลให้ระบบถูกโจมตีหรือควบคุมได้โดยสมบูรณ์

3.4.2.2 High ช่องโหว่ที่มีความเสี่ยงสูงและอาจถูกใช้ในการโจมตีระบบได้ง่าย

3.4.2.3 Medium ช่องโหว่ที่มีความเสี่ยงปานกลาง ต้องการเงื่อนไขเพิ่มเติมในการโจมตี

3.4.2.4 Low ช่องโหว่ที่มีความเสี่ยงต่ำ แต่ยังคงเป็นข้อบกพร่องที่ควรได้รับการแก้ไข

3.4.2.5 Informational ข้อมูลที่เกี่ยวข้องกับความมั่นคงปลอดภัยแต่ไม่ใช่ช่องโหว่โดยตรง

3.4.3 การดำเนินการทดสอบ

กระบวนการทดสอบช่องโหว่ดำเนินการเป็น 2 รอบ โดยรอบแรกเป็นการสแกนช่องโหว่ก่อนการแก้ไข และรอบที่สองเป็นการทดสอบหลังการปรับปรุงระบบ ดังนี้

3.4.3.1 สแกนช่องโหว่ก่อนการแก้ไข โดยใช้ OWASP ZAP, OpenVAS และ Nessus ทำการสแกนช่องโหว่เริ่มต้น ของระบบ Moodle 3.7.1 และ 4.5.1 เพื่อวิเคราะห์ และบันทึกจำนวนช่องโหว่ที่พบ แบ่งตามระดับ Critical, High, Medium, Low และ Informational และรวบรวมข้อมูลเพื่อนำไปใช้ในการปรับปรุงระบบ

3.4.3.2 แก้ไขและปรับปรุงระบบ ดำเนินการแก้ไขช่องโหว่ที่พบตามแนวทางความมั่นคงปลอดภัย เช่น การใช้ Prepared Statements การตั้งค่า Content Security Policy (CSP)

การบังคับใช้ SSL/TLS การจำกัดสิทธิ์ผู้ใช้ (RBAC) ปรับปรุงการตั้งค่าของ Apache/Nginx, MariaDB ให้เป็นไปตามมาตรฐานความมั่นคงปลอดภัย

3.4.3.3 สแกนช่องโหว่อีกครั้งหลังการแก้ไข ใช้ OWASP ZAP, OpenVAS และ Nessus ทำการ สแกนซ้ำ เพื่อตรวจสอบว่ายังพบช่องโหว่อยู่หรือไม่ เปรียบเทียบผลลัพธ์ก่อน และหลังการแก้ไข เพื่อประเมินประสิทธิภาพของมาตรการป้องกันที่นำมาใช้ วิเคราะห์ว่าช่องโหว่ประเภทใดยังคงมีอยู่ และหากยังพบช่องโหว่ที่มีความร้ายแรง ให้ดำเนินการแก้ไขเพิ่มเติม

บทที่ 4
ผลการดำเนินการวิจัย

ในการประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์ Moodle ได้ทำการสแกนช่องโหว่โดยใช้เครื่องมือ 3 ประเภท ได้แก่ OWASP ZAP, OpenVAS และ Nessus สำหรับ Moodle 3.7.1 และ Moodle 4.5.1 การสแกนดำเนินการทั้งก่อนและหลังการแก้ไข เพื่อเปรียบเทียบความแตกต่างของช่องโหว่ที่พบ และประเมินประสิทธิภาพของการปรับปรุงความมั่นคงปลอดภัย โดยแบ่งผลการดำเนินการเป็น 6 ส่วนดังนี้

4.1 ผลการสแกนช่องโหว่ก่อนการแก้ไข

4.2 การแก้ไขช่องโหว่

4.3 ผลการสแกนช่องโหว่หลังการแก้ไข

4.4 การจัดกลุ่มประเภทของช่องโหว่

4.5 การประเมินงบประมาณในการอัปเกรดระบบ

4.6 การวิเคราะห์ความเสี่ยงของช่องโหว่ที่ยังเหลืออยู่

4.1 ผลการสแกนช่องโหว่ก่อนการแก้ไข (Initial Vulnerability Scan Results)

เพื่อประเมินความมั่นคงปลอดภัยของ Moodle 3.7.1 และ Moodle 4.5.1 ได้ดำเนินการสแกนช่องโหว่ก่อนการแก้ไขโดยใช้ OWASP ZAP, OpenVAS และ Nessus เพื่อตรวจสอบความเสี่ยงด้านความมั่นคงปลอดภัยในระดับต่างๆ

ผลลัพธ์ก่อนการแก้ไขสรุปและเปรียบเทียบจำนวนช่องโหว่ของแต่ละเครื่องมือไว้ใน ตารางที่ 4-1 โดยรายละเอียดช่องโหว่ของ Moodle 3.7.1 แสดงใน ตารางที่ ก-1 ข-1 และ ค-1 และของ Moodle 4.5.1 แสดงใน ตารางที่ ก-2 ข-2 และ ค-2

จากผลการวิเคราะห์ ได้ดำเนินการแก้ไขช่องโหว่หลายส่วนเพื่อเพิ่มความมั่นคงปลอดภัยของระบบ โดยมีรายละเอียดดังนี้

ตารางที่ 4-1 แสดงการเปรียบเทียบจำนวนช่องโหว่ที่ตรวจพบในแต่ละประเภทของเครื่องมือสแกนก่อนการแก้ไขช่องโหว่

NO.	CATEGORY	NESSUS		OWASP ZAP		OPENVAS	
		Moodle 3.7.1	Moodle 4.5.1	Moodle 3.7.1	Moodle 4.5.1	Moodle 3.7.1	Moodle 4.5.1
1	Critical	30	0	0	0	0	0
2	High	31	0	1	1	6	1
3	Medium	23	2	4	2	1	1
4	Low	2	0	7	5	0	0
5	Info	42	39	7	7	0	0

จากตารางที่ 4-1 แสดงผลการเปรียบเทียบจำนวนช่องโหว่ก่อนการแก้ไข พบว่าเครื่องมือ Nessus ตรวจพบช่องโหว่ใน Moodle 3.7.1 เป็นจำนวนมาก โดยเฉพาะในระดับ Critical จำนวน 30 รายการ ระดับ High จำนวน 31 รายการ และระดับ Medium อีก 23 รายการ ซึ่งถือว่ามีความรุนแรงในระดับสูงถึงปานกลางเป็นจำนวนมาก เมื่อเปรียบเทียบกับเครื่องมือ OWASP ZAP และ OpenVAS ที่ไม่สามารถตรวจพบช่องโหว่ระดับ Critical ได้เลยในทั้งสองเวอร์ชัน เครื่องมือ Nessus มีความสามารถในการวิเคราะห์ช่องโหว่ด้านการตั้งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ (Server Misconfiguration) และระบบปฏิบัติการได้อย่างครอบคลุม จึงสามารถตรวจพบช่องโหว่ได้มากกว่าเครื่องมืออื่น ในขณะที่ OWASP ZAP เน้นการตรวจสอบช่องโหว่ในระดับ Web Application เช่น SQL Injection และ Cross-Site Scripting (XSS) ซึ่งพบในระดับ Medium และ Low เพียงเล็กน้อย OpenVAS ตรวจพบช่องโหว่ในระดับ High จำนวน 6 รายการใน Moodle 3.7.1 และ 1 รายการใน Moodle 4.5.1 แสดงถึงขีดความสามารถในการตรวจสอบความมั่นคงปลอดภัยของระบบเครือข่ายและบริการระบบปฏิบัติการที่มีความลึกน้อยกว่า Nessus

นอกจากนี้จากการเปรียบเทียบระหว่าง Moodle 3.7.1 และ Moodle 4.5.1 พบว่าจำนวนช่องโหว่ลดลงอย่างมีนัยสำคัญในทุกเครื่องมือ โดยเฉพาะ Nessus ที่ไม่พบช่องโหว่ระดับ Critical และ High ใน Moodle 4.5.1 เลย ซึ่งแสดงให้เห็นว่ามีการปรับปรุงความมั่นคงปลอดภัยของระบบในเวอร์ชันใหม่ได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ยังพบช่องโหว่ในระดับ Medium และ Low อยู่ในบางเครื่องมือ โดยเฉพาะ OWASP ZAP ที่ยังตรวจพบ XSS และปัญหาด้าน Header Security ในระดับความรุนแรงปานกลาง ซึ่งควรได้รับการตรวจสอบเพิ่มเติมในกระบวนการปรับปรุงระบบต่อไป

4.2 การแก้ไขช่องโหว่

ก่อนดำเนินการแก้ไขช่องโหว่ ระบบได้มีการวิเคราะห์และคัดเลือกช่องโหว่ที่มีระดับความรุนแรงสูงและมีผลกระทบโดยตรงต่อความมั่นคงปลอดภัยของระบบ Moodle โดยพิจารณาจากช่องโหว่ที่ตรวจพบจากการสแกนด้วยเครื่องมือ Nessus, OWASP ZAP และ OpenVAS โดยเฉพาะช่องโหว่ที่อยู่ในระดับ Critical, High และ Medium ตามมาตรฐานของ CVSS (Common Vulnerability Scoring System) ช่องโหว่ที่ถูกเลือกมาแก้ไข ได้แก่ ช่องโหว่ด้านการตรวจสอบอินพุต (Input Validation Issues) ช่องโหว่จากการตั้งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ (Security Misconfiguration) และช่องโหว่จากการอัปโหลดไฟล์ที่ไม่ปลอดภัย ซึ่งส่งผลโดยตรงต่อการเข้าถึงระบบฐานข้อมูลหรือการควบคุมระบบจากภายนอก

เครื่องมือที่ตรวจพบช่องโหว่ดังกล่าว ได้แก่ Nessus ซึ่งตรวจพบช่องโหว่ระดับ High และ Medium ในหลายรายการ โดยเฉพาะช่องโหว่ที่เกี่ยวข้องกับการตั้งค่าระบบที่ไม่ปลอดภัย การจัดการสิทธิ์ที่ไม่เหมาะสม และปัญหาเกี่ยวกับ SSL/TLS ขณะที่ OWASP ZAP ตรวจพบช่องโหว่ในระดับ Medium ที่เกี่ยวข้องกับ Cross-Site Scripting (XSS) และการจัดการ Input Validation ที่ไม่เพียงพอ การเลือกช่องโหว่เหล่านี้มาจัดการในลำดับแรก เนื่องจากมีโอกาสถูกโจมตีสูง และอาจส่งผลกระทบต่อความมั่นคงปลอดภัยของผู้ใช้งานและข้อมูลของวิทยาลัยการศึกษาโดยตรง แนวทางการแก้ไขแต่ละช่องโหว่ มีรายละเอียดดังนี้

4.2.1 การแก้ไขช่องโหว่ด้านการตรวจสอบอินพุต (Input Validation Issues)

เครื่องมือที่ตรวจพบ: Nessus (High, Medium), OWASP ZAP (Medium)

ช่องโหว่ด้านการตรวจสอบอินพุตตรวจพบโดยเครื่องมือ Nessus และ ZAP โดยอยู่ในระดับความรุนแรงตั้งแต่ปานกลางถึงสูง ช่องโหว่เหล่านี้เกี่ยวข้องกับการรับค่าข้อมูลจากผู้ใช้งานโดยไม่มีการกรองหรือยืนยันความถูกต้องอย่างเพียงพอ ซึ่งอาจเปิดโอกาสให้ผู้ไม่ประสงค์ดีสามารถส่งโค้ดที่เป็นอันตราย เช่น คำสั่ง SQL หรือ JavaScript เข้าสู่ระบบได้

แนวทางการแก้ไข: ป้องกัน SQL Injection โดยการใช้ Prepared Statements และ Parameter Binding ในการสื่อสารกับฐานข้อมูล ปิดการแสดงข้อผิดพลาดฐานข้อมูล (display_errors = Off) เพื่อลดความเสี่ยงจากข้อมูลที่อาจถูกเปิดเผย

4.2.2 การแก้ไขช่องโหว่ด้านการตั้งค่า Header Security (Security Header Misconfiguration)

เครื่องมือที่ตรวจพบ: OWASP ZAP (Medium), Nessus (Medium)

จากการตรวจสอบพบว่าเครื่องมือ OWASP ZAP และ Nessus ตรวจพบช่องโหว่ที่เกี่ยวข้องกับการตั้งค่า HTTP Response Header ที่ไม่เหมาะสมหรือไม่มีการกำหนดค่าเพื่อป้องกันการโจมตีในระดับ Web Application โดยเฉพาะในส่วนของการป้องกัน Cross-Site Scripting (XSS), Session Hijacking และ Clickjacking ซึ่งเป็นช่องโหว่ที่มีระดับความรุนแรงปานกลาง (Medium) และพบได้ในทั้ง Moodle 3.7.1 และ 4.5.1 ก่อนการปรับปรุง

แนวทางการแก้ไข: เพิ่มการตั้งค่าหัวข้อ Content Security Policy (CSP) เพื่อจำกัดแหล่งที่มาของสคริปต์และทรัพยากรภายนอก ซึ่งช่วยลดความเสี่ยงจากการโจมตีแบบ XSS พร้อมทั้งกำหนดค่า HttpOnly, Secure และ SameSite Attributes บนคุกกี้ เพื่อป้องกันการดักจับ Session และการโจมตีแบบ Session Hijacking รวมถึงการตั้งค่า X-Frame-Options เป็น SAMEORIGIN เพื่อป้องกันการฝังหน้าเว็บจากภายนอก (Clickjacking Attack) โดยการตั้งค่าทั้งหมดนี้ได้ดำเนินการผ่านไฟล์คอนฟิกของ Web Server (เช่น NGINX หรือ Apache) และได้รับการยืนยันผลการแก้ไขผ่านการทดสอบซ้ำโดยเครื่องมือ OWASP ZAP แล้วไม่พบช่องโหว่ดังกล่าวอีก

4.2.3 การตั้งค่า SSL/TLS และการใช้งาน DuckDNS เพื่อเพิ่มความมั่นคงปลอดภัยของเซิร์ฟเวอร์

เครื่องมือที่ตรวจพบ: Nessus (Medium), OWASP ZAP (Medium)

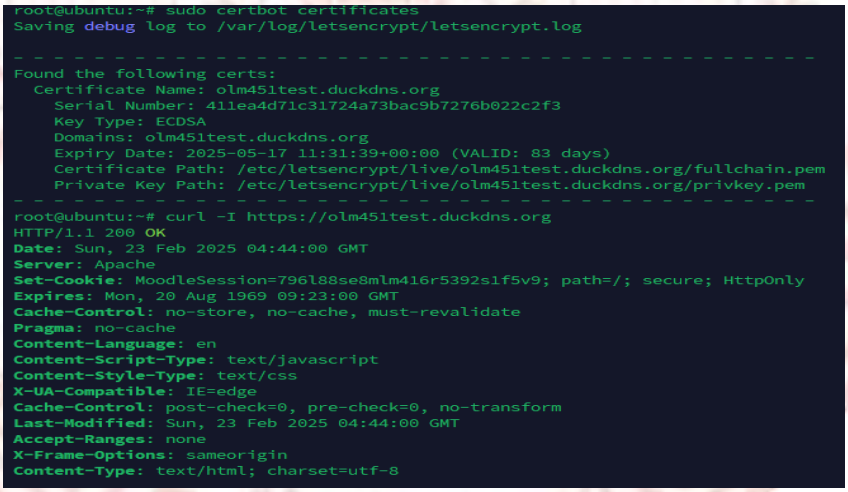
จากการตรวจสอบพบว่าเครื่องมือ Nessus และ OWASP ZAP ตรวจพบช่องโหว่ด้านความมั่นคงปลอดภัยที่เกี่ยวข้องกับการไม่ใช้งาน SSL/TLS หรือการตั้งค่าที่ไม่ปลอดภัย เช่น การอนุญาตให้เข้าถึงเว็บไซต์ผ่าน HTTP ที่ไม่มีการเข้ารหัส การใช้ใบรับรองที่หมดอายุหรือไม่เชื่อถือได้ และไม่มีการบังคับใช้การเชื่อมต่อผ่าน HTTPS อย่างสมบูรณ์ ช่องโหว่เหล่านี้พบได้ในระดับความรุนแรงปานกลาง (Medium) และปรากฏในทั้ง Moodle 3.7.1 และ 4.5.1 ก่อนดำเนินการปรับปรุง

แนวทางการแก้ไข: ระบบได้ดำเนินการตั้งค่าให้รองรับ SSL/TLS อย่างสมบูรณ์โดยใช้ใบรับรองความมั่นคงปลอดภัยจาก Let’s Encrypt ซึ่งเป็น Certificate Authority ที่น่าเชื่อถือ และมีการต่ออายุอัตโนมัติผ่าน certbot เพื่อลดความเสี่ยงจากใบรับรองหมดอายุ พร้อมทั้งกำหนดให้บังคับใช้งานเว็บไซต์ผ่าน HTTPS เท่านั้น นอกจากนี้ยังได้ใช้บริการ DuckDNS เพื่อกำหนดชื่อโดเมนแบบ Dynamic DNS ให้

สามารถเข้าถึงเซิร์ฟเวอร์ได้อย่างปลอดภัย และมีเสถียรภาพมากยิ่งขึ้น การตั้งค่าเหล่านี้ช่วยให้การเข้าถึงเว็บไซต์มีความมั่นคงปลอดภัยจากการดักจับข้อมูล (Man-in-the-Middle Attack) และลดความเสี่ยงจากการโจมตีในระดับ Transport Layer

หลังจากดำเนินการติดตั้งและกำหนดค่าเสร็จสิ้น ได้มีการตรวจสอบความถูกต้องของใบรับรอง SSL และความสามารถในการเข้าถึง Moodle ผ่าน HTTPS ซึ่งแสดงใน ภาพที่ 4-1 และ ภาพที่ 4-2

ภาพที่ 4-1 ผลลัพธ์จากคำสั่งที่ใช้ตรวจสอบ ใบรับรอง SSL เมื่อเข้าถึง Moodle 3.7.1



ภาพที่ 4-2 ผลลัพธ์จากคำสั่งที่ใช้ตรวจสอบ ใบรับรอง SSL เมื่อเข้าถึง Moodle 4.5.1

การตั้งค่า SSL/TLS ช่วยให้ Moodle สามารถทำงานภายใต้โปรโตคอลที่ปลอดภัย ลดความเสี่ยงจากการดักจับข้อมูล และเป็นไปตามแนวทางปฏิบัติด้านความมั่นคงปลอดภัยของเว็บแอปพลิเคชันที่ทันสมัย

4.3 ผลการสแกนช่องโหว่หลังการแก้ไข (Post-Mitigation Vulnerability Scan Results)

หลังจากดำเนินการปรับปรุงระบบและแก้ไขช่องโหว่ในประเด็นต่าง ๆ ที่ตรวจพบก่อนหน้านี้ ผู้วิจัยได้ดำเนินการสแกนซ้ำ (Re-assessment) โดยใช้เครื่องมือ Nessus, OWASP ZAP และ

OpenVAS เพื่อประเมินผลลัพธ์หลังการปรับปรุง พบว่าจำนวนช่องโหว่ที่ตรวจพบโดยเฉพาะในระดับความรุนแรงสูง ได้แก่ Critical และ High ลดลงอย่างมีนัยสำคัญ โดยใน Moodle 4.5.1 ไม่ตรวจพบช่องโหว่ในสองระดับนี้จากทุกเครื่องมือเลย ขณะที่ Moodle 3.7.1 ยังตรวจพบช่องโหว่ระดับ Medium และ Low โดยเฉพาะจาก Nessus และ OWASP ZAP แม้ว่าจำนวนจะลดลงเมื่อเทียบกับก่อนการแก้ไขก็ตาม

ผลลัพธ์ก่อนการแก้ไขได้ถูกสรุปและเปรียบเทียบจำนวนช่องโหว่ที่ตรวจพบจากแต่ละเครื่องมือไว้ใน ตารางที่ 4-2 โดยรายละเอียดของช่องโหว่สำหรับ Moodle 3.7.1 สามารถดูได้ใน ตารางที่ ก-2 ข-2 และ ค-2 และสำหรับ Moodle 4.5.1 สามารถดูได้ใน ตารางที่ ก-4 ข-4 และ ค-4

ตารางที่ 4-2 แสดงการเปรียบเทียบจำนวนช่องโหว่ที่ตรวจพบในแต่ละประเภทของเครื่องมือสแกนหลังการแก้ไขช่องโหว่

NO.	CATEGORY	NESSUS		OWASP ZAP		OPENVAS	
		Moodle 3.7.1	Moodle 4.5.1	Moodle 3.7.1	Moodle 4.5.1	Moodle 3.7.1	Moodle 4.5.1
1	Critical	15	0	0	0	0	0
2	High	15	0	1	0	0	0
3	Medium	11	0	5	2	2	0
4	Low	1	0	8	7	1	1
5	Info	37	35	9	8	0	0

ตารางที่ 4-2 แสดงให้เห็นว่าหลังจากดำเนินการแก้ไขช่องโหว่ตามมาตรการด้านความมั่นคงปลอดภัยแล้ว จำนวนช่องโหว่ที่ตรวจพบใน Moodle 4.5.1 ลดลงอย่างชัดเจนในทุกเครื่องมือ โดยไม่พบช่องโหว่ระดับ Critical และ High อีกเลย ซึ่งแตกต่างจาก Moodle 3.7.1 ที่ยังตรวจพบช่องโหว่ระดับ Critical และ High จากเครื่องมือ Nessus อย่างละ 15 รายการ แสดงให้เห็นว่าการปรับปรุงเวอร์ชันและการตั้งค่าความมั่นคงปลอดภัยใหม่สามารถลดความเสี่ยงที่มีความรุนแรงสูงได้อย่างมีประสิทธิภาพ

เมื่อพิจารณาช่องโหว่ที่ยังหลงเหลือหลังการแก้ไข พบว่ามีช่องโหว่ระดับ Medium และ Low ที่ยังคงตรวจพบโดย OWASP ZAP และ Nessus ใน Moodle ทั้งสองเวอร์ชัน โดย Moodle 3.7.1 ยังคงมีช่องโหว่ระดับ Medium จาก Nessus จำนวน 11 รายการ และจาก ZAP จำนวน 5 รายการ ในขณะที่ Moodle 4.5.1 ยังมีช่องโหว่ระดับ Medium จาก ZAP จำนวน 2 รายการ ซึ่งส่วนใหญ่อยู่ในกลุ่มช่องโหว่ด้าน Input Validation, Header Security และการตั้งค่าคุกกี้ (Cookie Attributes) ที่ยังไม่ครบถ้วนตามแนวทางความมั่นคงปลอดภัย เช่น ยังไม่มีการตั้งค่า SameSite, HttpOnly หรือ Secure กับทุก Session Cookie รวมถึงการขาด Content Security Policy (CSP) ที่ครอบคลุมทุกหน้าเว็บ

ในส่วนของ OpenVAS ไม่พบช่องโหว่ระดับ Critical, High หรือ Medium ในทั้งสองเวอร์ชัน ซึ่งอาจสะท้อนถึงข้อจำกัดของเครื่องมือที่เน้นการตรวจสอบในระดับระบบเครือข่ายหรือเซิร์ฟเวอร์มากกว่าระดับ Web Application ที่ Moodle ทำงานอยู่

สำหรับช่องโหว่บางรายการที่ยังคงหลงเหลืออยู่และไม่สามารถดำเนินการแก้ไขได้ในขอบเขตของการศึกษาครั้งนี้ ได้แก่ ช่องโหว่ที่เกิดจากปลั๊กอินของ Moodle ซึ่งยังไม่มีเวอร์ชันที่อัปเดตเพื่อแก้ไขช่องโหว่ หรือไม่สามารถปรับแต่งได้หากไม่กระทบกับฟังก์ชันการทำงานหลักของระบบ เช่น การแสดงผล JavaScript ของบางโมดูลที่จำเป็นต้องใช้ inline script ซึ่งขัดกับการตั้งค่า CSP รวมถึงข้อจำกัดด้านเวลาที่ทำให้ไม่สามารถเขียนโค้ดแก้ไขปลั๊กอินใหม่ได้ จึงต้องยอมรับความเสี่ยงระดับต่ำในบางจุด พร้อมกำหนดมาตรการควบคุมเชิงนโยบาย เช่น การควบคุมสิทธิ์ผู้ใช้ และการจำกัดการเข้าถึง

อย่างไรก็ตาม หลังจากดำเนินการแก้ไขช่องโหว่ด้านความมั่นคงปลอดภัยแล้ว จำเป็นต้องทดสอบการทำงานของระบบเพื่อให้แน่ใจว่าการแก้ไข ไม่ส่งผลกระทบต่อฟังก์ชันหลักของ Moodle และยังคงทำงานได้ตามปกติ โดยการทดสอบแบ่งออกเป็น 3 ส่วนหลัก ได้แก่

4.3.1 การทดสอบความมั่นคงปลอดภัย (Security Testing)

4.3.1.1 ตรวจสอบว่าการเปิดใช้งาน HTTPS & SSL สามารถเข้ารหัสข้อมูลได้ถูกต้อง

4.3.1.2 ตรวจสอบว่า Directory Listing ถูกปิดและไม่สามารถเข้าถึงไฟล์ที่ไม่จำเป็นได้

4.3.1.3 ตรวจสอบการอัปเดตซอฟต์แวร์ เช่น Moodle, PHP, Apache/Nginx, MariaDB ว่าทำงานได้ปกติ

4.3.2 การทดสอบฟังก์ชันพื้นฐานของระบบ (Core Functionality Testing)

4.3.2.1 ทดสอบการ ล็อกอิน และการลงทะเบียนผู้ใช้

4.3.2.2 ตรวจสอบการ สร้าง และเข้าถึงคอร์สเรียน

4.3.2.3 ทดสอบการอัปโหลดไฟล์ และการทำงานของ Plugin ต่าง ๆ

4.3.3 การทดสอบประสิทธิภาพ (Performance Testing)

4.3.3.1 ตรวจสอบความเร็วในการโหลดหน้าเว็บ หลังจากอัปเดต และแก้ไข

4.3.3.2 ทดสอบการเข้าใช้งานพร้อมกัน (Concurrent Users Test)

4.3.3.3 ตรวจสอบการทำงานของ Database และ Server Logs เพื่อดูว่ามี Error หรือไม่

หลังจากดำเนินการแก้ไขช่องโหว่ ระบบได้ผ่านการทดสอบใน 3 ด้านหลัก ได้แก่ ความมั่นคงปลอดภัย ฟังก์ชันการทำงานพื้นฐาน และประสิทธิภาพของระบบ โดยในด้านความมั่นคงปลอดภัย พบว่าระบบสามารถใช้งานผ่าน HTTPS ได้อย่างถูกต้อง พร้อมเปิดใช้งาน SSL และได้รับใบรับรองจาก Let's Encrypt ครบถ้วน ตรวจสอบแล้วไม่มีการเปิด Directory Listing และไม่สามารถเข้าถึงไฟล์ภายในที่ควรถูกป้องกัน เช่น ไฟล์ตั้งค่าระบบหรือฐานข้อมูล นอกจากนี้ยังตรวจสอบเวอร์ชันของซอฟต์แวร์ต่าง ๆ เช่น Moodle, PHP, Apache/Nginx และ MariaDB พบว่าอยู่ในเวอร์ชันที่มีความมั่นคงปลอดภัยหรือได้รับการอัปเดตแล้ว ด้านฟังก์ชันการทำงานพื้นฐาน ระบบสามารถเข้าสู่ระบบ ลงทะเบียนผู้ใช้งานใหม่ และสร้างคอร์สเรียนได้อย่างสมบูรณ์ การอัปโหลดไฟล์ การใช้งานปลั๊กอิน เช่น Quiz หรือ Attendance

สามารถทำงานได้อย่างถูกต้องและไม่มีข้อผิดพลาด สำหรับด้านประสิทธิภาพ พบว่าความเร็วในการโหลดหน้าเว็บไซต์ดีขึ้น โดยหน้าแรกโหลดได้ภายใน 2 วินาทีจากการปรับแต่งเซิร์ฟเวอร์ และจากการทดสอบการใช้งานพร้อมกันด้วย Apache Bench พบว่าสามารถรองรับผู้ใช้งานประมาณ 30–40 คนได้โดยไม่เกิดความล่าช้าหรือข้อผิดพลาด อีกทั้งจากการตรวจสอบ Server Logs และฐานข้อมูล ไม่พบ Error ที่มีความรุนแรงหรือส่งผลกระทบต่อระบบ ทำให้สามารถสรุปได้ว่าระบบมีความมั่นคงปลอดภัย และพร้อมใช้งานในสภาพแวดล้อมจริง

4.4 การจัดกลุ่มประเภทของช่องโหว่ (Vulnerability Grouping)

ผลการประเมินช่องโหว่ของระบบ Moodle ทั้งเวอร์ชัน 3.7.1 และ 4.5.1 โดยใช้เครื่องมือสแกนช่องโหว่มาตรฐาน ได้แก่ OWASP ZAP, Nessus และ OpenVAS พบว่าช่องโหว่ที่ตรวจพบมีความหลากหลายทั้งในด้านประเภทและระดับความรุนแรง เพื่อให้การวิเคราะห์มีความชัดเจนยิ่งขึ้น จึงได้มีการจัดกลุ่มช่องโหว่ตามลักษณะร่วม (Vulnerability Grouping) ซึ่งอ้างอิงจากแนวทางของ OWASP Top 10 และ CVSS (Common Vulnerability Scoring System)

การจัดกลุ่มนี้ช่วยให้สามารถวางมาตรการป้องกันเชิงระบบได้อย่างมีประสิทธิภาพ และช่วยให้เห็นแนวโน้มของปัญหาด้านความมั่นคงปลอดภัยที่เกิดขึ้นซ้ำในลักษณะเดียวกัน โดยในตารางที่ 4-3 แสดงการจัดกลุ่มประเภทของช่องโหว่ จำนวนที่ตรวจพบ รายการช่องโหว่ เครื่องมือที่ตรวจพบ และระดับความรุนแรงที่อ้างอิงจากผลการสแกนของแต่ละเครื่องมือ

ตารางที่ 4-3 แสดงการการจัดกลุ่มช่องโหว่ตามประเภท

NO.	ประเภทช่องโหว่	รายการช่องโหว่	จำนวนที่พบ (รวม)	เครื่องมือที่พบ	PLUGIN ID
1	Input Validation	XSS, SQL Injection	22	ZAP, Nessus	ZAP: 10038 / Nessus: 45590
2	Server Misconfiguration	ไม่มีการตั้งค่า SSL / Header	45	Nessus, ZAP	ZAP: 10038
3	TLS/HTTPS Issues	ไม่มี HTTPS, TLS v1.0	12	Nessus	Nessus: 104743
4	Insecure Cookie / Session	ไม่มี HttpOnly, SameSite	10	ZAP	ZAP: 10010, 10054
5	Cryptographic Weakness	Weak MAC Algorithms, SHA-1 Signed Cert	2	OpenVAS	OpenVAS: CWE-327
6	Outdated Components	Plugin Moodle เก่า	5	Nessus, OpenVAS	Nessus: 152853
7	Information Disclosure	Timestamp Disclosure, Suspicious Comments	69	ZAP	Nessus: 152853 / OpenVAS: 1.3.5.4-1
8	Third-party Scripts	Cross-Domain JavaScript Inclusion	14	ZAP	ZAP: 10017

จากการจัดกลุ่มช่องโหว่ตามประเภทและระดับความรุนแรง พบว่า ช่องโหว่ส่วนใหญ่มีสาเหตุมาจากการกำหนดค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ที่ไม่เหมาะสม (Security Misconfiguration) และ

การตรวจสอบข้อมูลนำเข้าที่ไม่เพียงพอ (Input Validation) ซึ่งเป็นปัจจัยหลักที่นำไปสู่ความเสี่ยงด้านความมั่นคงปลอดภัยในระบบ Moodle โดยเฉพาะอย่างยิ่งช่องโหว่ที่เกี่ยวข้องกับ XSS การตั้งค่า Header และการใช้โปรโตคอล TLS ที่ล้าสมัย

การวิเคราะห์นี้ช่วยให้สามารถมองเห็นแนวโน้มของช่องโหว่ที่เกิดขึ้นซ้ำในหลายเวอร์ชันของระบบ และสามารถนำไปสู่การวางมาตรการแก้ไขและป้องกันในลักษณะรวมกลุ่ม (bulk mitigation) ได้อย่างมีประสิทธิภาพ เช่น การตั้งค่าหัวข้อ HTTP Security Header อย่างครบถ้วน หรือการบังคับใช้ HTTPS และ TLS เวอร์ชันที่ปลอดภัย

นอกจากนี้ ข้อมูลในตารางยังสามารถนำไปใช้ในการจัดลำดับความสำคัญในการแก้ไขช่องโหว่ตามระดับความรุนแรง เพื่อให้การใช้ทรัพยากรในการดูแลรักษาระบบมีประสิทธิภาพสูงสุด

4.5 การประเมินงบประมาณในการอัปเกรดระบบ

ผลการวิเคราะห์ช่องโหว่ที่เหลืออยู่และข้อจำกัดเชิงเทคนิคจาก Moodle เวอร์ชัน 3.7.1 พบว่าแม้จะสามารถดำเนินการปรับปรุงระบบเพื่อเพิ่มความมั่นคงปลอดภัยได้ในระดับหนึ่ง แต่ยังคงมีข้อจำกัดหลายประการที่ไม่สามารถแก้ไขได้ในเวอร์ชันเดิม เช่น การสนับสนุนมาตรฐานด้านความมั่นคงปลอดภัยล่าสุด (TLS, CSP, RBAC) รวมถึงปลั๊กอินรุ่นเก่าที่หมดอายุการสนับสนุน (End of Life)

เพื่อให้สามารถเปรียบเทียบทางเลือกในการพัฒนาระบบได้อย่างเป็นระบบ จึงได้จัดทำการประเมินงบประมาณใน 2 แนวทาง ได้แก่

แนวทางที่ 1: การปรับปรุงระบบเดิมในเวอร์ชัน 3.7.1 โดยไม่เปลี่ยนเวอร์ชัน

แนวทางที่ 2: การอัปเกรดระบบเป็นเวอร์ชันใหม่ 4.5.1 พร้อมจัดหาเครื่องแม่ข่าย (Server)

โดยใช้อัตราค่าจ้างและต้นทุนที่อ้างอิงจากเกณฑ์ราคากลางของกรมบัญชีกลาง พ.ศ. 2567 [19] และราคากลางอุปกรณ์จากกระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม [20] รวมถึงแนวทางค่าตอบแทนตามระเบียบของวิทยาลัยที่ผู้วิจัยใช้เป็นกรณีศึกษา [21]

ตารางที่ 4-4 การปรับปรุงระบบเดิมใน Moodle 3.7.1 โดยไม่เปลี่ยนเวอร์ชัน

NO.	รายการดำเนินการ	เวลา (ชม.)	อัตรา (บ./ชม.)	รวม (บาท)	หมายเหตุ
1	สำรองข้อมูลก่อนการอัปเกรดระบบ	2	450	900	จำเป็นต้องสำรองก่อนดำเนินการอัปเกรด
2	ปิด Weak MAC Algorithm ใน SSH	2	450	900	ลดความเสี่ยงจากการโจมตีผ่าน SSH (อ้างอิง CIS Benchmark)
3	ติดตั้งระบบสำรองข้อมูลอัตโนมัติ	4	450	1,800	เพื่อการกู้คืนระบบในกรณีข้อมูลเสียหาย
4	ติดตั้ง SSL/TLS	2	450	900	อิงราคาค่าจ้างจากกรมบัญชีกลาง (ป.โท 10 ปี = 450 บ./ชม.)

ตารางที่ 4-4 (ต่อ)

NO.	รายการดำเนินการ	เวลา (ชม.)	อัตรา (บ./ชม.)	รวม (บาท)	หมายเหตุ
5	ตั้งค่า Cookie Attributes (HttpOnly, Secure, SameSite)	2	450	900	ลดช่องโหว่ Session Hijacking (ตาม OWASP)
6	กำหนดการควบคุมสิทธิ์ RBAC	3	450	1,350	เพิ่มความมั่นคงปลอดภัยในการเข้าถึงระบบ
7	ตรวจสอบช่องโหว่ด้วย OWASP ZAP / Nessus / OpenVAS	6	450	2,700	ใช้เครื่องมือสแกนช่องโหว่ตามแนวทาง OWASP
8	ทดสอบระบบบน Staging Environment	6	450	2,700	ใช้ก่อน Deploy ระบบจริง
	รวมงบประมาณ			**12,150**	

จากการวิเคราะห์พบว่า การปรับปรุงระบบใน Moodle 3.7.1 โดยไม่เปลี่ยนเวอร์ชัน มีค่าใช้จ่ายรวมประมาณ 12,150 บาท ซึ่งครอบคลุมถึงการตั้งค่าความมั่นคงปลอดภัยใหม่ (Header, Cookie, RBAC) การทดสอบช่องโหว่ อย่างไรก็ตาม ระบบเวอร์ชันเดิมยังคงมีข้อจำกัดในด้านความเข้ากันได้กับมาตรฐานความมั่นคงปลอดภัยสมัยใหม่ และมีแนวโน้มจะขาดการสนับสนุนในอนาคต

ตารางที่ 4-5 การอัปเกรดระบบเป็นเวอร์ชันใหม่ 4.5.1 พร้อมจัดหาเครื่องแม่ข่าย (Server)

NO.	รายการดำเนินการ	เวลา (ชม.)	อัตรา (บ./ชม.)	รวม (บาท)	หมายเหตุ
1	สำรองข้อมูลก่อนการอัปเกรดระบบ	2	450	900	จำเป็นต้องสำรองก่อนดำเนินการอัปเกรด
2	ย้ายข้อมูลจากระบบเดิม (Data Migration)	6	450	2,700	ใช้เวลาเฉลี่ย 6 ชั่วโมง ตามขั้นตอน migration
3	จัดหา/ติดตั้ง Server รองรับ Moodle 4.5.1	-	-	130,000	ราคากลางเครื่องแม่ข่ายแบบที่ 1 จากกระทรวงดิจิทัลฯ (MDES)
4	ปิด Weak MAC Algorithm ใน SSH	2	450	900	ลดความเสี่ยงจากการโจมตีผ่าน SSH (อ้างอิง CIS Benchmark)
5	ติดตั้งระบบสำรองข้อมูลอัตโนมัติ	4	450	1,800	เพื่อการกู้คืนระบบในกรณีข้อมูลเสียหาย
6	ติดตั้ง SSL/TLS	2	450	900	อิงราคาค่าจ้างจากกรมบัญชีกลาง (ป.โท 10 ปี = 450 บ./ชม.)
7	ตั้งค่า Content Security Policy (CSP)	3	450	1,350	อิงมาตรฐาน OWASP Security Header

ตารางที่ 4-5 (ต่อ)

NO.	รายการดำเนินการ	เวลา (ชม.)	อัตรา (บ./ชม.)	รวม (บาท)	หมายเหตุ
8	ตั้งค่า Cookie Attributes (HttpOnly, Secure, SameSite)	2	450	900	ลดช่องโหว่ Session Hijacking (ตาม OWASP)
9	กำหนดการควบคุมสิทธิ์ RBAC	3	450	1,350	เพิ่มความมั่นคงปลอดภัยในการเข้าถึงระบบ
10	ตรวจสอบช่องโหว่ด้วย OWASP ZAP / Nessus / OpenVAS	6	450	2,700	ใช้เครื่องมือสแกนช่องโหว่ตามแนวทาง OWASP
11	ทดสอบระบบบน Staging Environment	6	450	2,700	ใช้ก่อน Deploy ระบบจริง
12	อบรมอาจารย์ผู้ใช้ระบบ (70 คน)	-	-	10,500	ค่าอาหารว่าง/เครื่องดื่ม คนละ 150 บาท
13	ค่าจ้างวิทยากรอบรม (1 วัน)	-	-	15,000	อัตราวิทยากรกลาง 1 วัน (วิทยาลัยของผู้วิจัย)
14	ค่าจ้างที่ปรึกษา (1 เดือน) Business Analyst (BA)	-	-	70,050	อัตรารายเดือนของกรมบัญชีกลาง
15	ค่าจ้างที่ปรึกษา (1 เดือน) System Analyst (SA)	-	-	70,050	อัตรารายเดือนของกรมบัญชีกลาง
16	ค่าจ้างที่ปรึกษา (1 เดือน) Developer (Dev)	-	-	70,050	อัตรารายเดือนของกรมบัญชีกลาง
	รวมงบประมาณ			**381,850**	

สำหรับการอัปเกรดระบบเป็น Moodle 4.5.1 พร้อมจัดหาเครื่องแม่ข่ายใหม่ มีค่าใช้จ่ายรวมประมาณ 381,850 บาท โดยรวมถึงค่าใช้จ่ายในขั้นตอนการย้ายข้อมูล (Data Migration) การจัดหาอุปกรณ์ การติดตั้งระบบ การกำหนดค่าความมั่นคงปลอดภัย การทดสอบระบบ และการฝึกอบรมครบถ้วน ซึ่งช่วยยกระดับความมั่นคงปลอดภัยของระบบให้เป็นไปตามมาตรฐานสมัยใหม่ เช่น OWASP, TLS 1.2+, CSP และการควบคุมสิทธิ์ (RBAC)

แม้มีค่าใช้จ่ายสูงกว่าแนวทางการปรับปรุงเวอร์ชันเดิม แต่การอัปเกรดจะช่วยลดความเสี่ยงด้านความมั่นคงปลอดภัยในระยะยาว และสามารถรองรับการพัฒนาระบบหรือขยายตัวในอนาคตได้อย่างมีประสิทธิภาพและยั่งยืน

4.6 การวิเคราะห์ความเสี่ยงของช่องโหว่ที่ยังเหลืออยู่

หลังจากที่ได้ดำเนินการปรับปรุงระบบ Moodle และอัปเกรดเวอร์ชัน พร้อมทั้งกำหนดมาตรการด้านความมั่นคงปลอดภัยตามแนวทางที่วางไว้แล้ว ได้มีการสแกนระบบอีกครั้งโดยใช้เครื่องมือ Nessus,

OWASP ZAP และ OpenVAS เพื่อประเมินช่องโหว่ที่ยังคงปรากฏอยู่ ผลการสแกนดังแสดงในตารางที่ 4-2 พบว่า ระบบ Moodle เวอร์ชัน 4.5.1 ที่ได้รับการอัปเกรด ไม่พบช่องโหว่ในระดับ Critical และ High อีกเลย แสดงให้เห็นถึงประสิทธิภาพของการอัปเกรดในการลดความเสี่ยงเชิงระบบ ในขณะที่ระบบ Moodle เวอร์ชัน 3.7.1 ที่ได้รับการ “ปรับปรุงเฉพาะจุด” ยังคงปรากฏช่องโหว่ระดับสูงอยู่หลายรายการในทุกเครื่องมือ

สำหรับช่องโหว่ที่ยังคงเหลืออยู่ในระบบ Moodle 4.5.1 ส่วนใหญ่เป็นช่องโหว่ระดับต่ำ (Low) และข้อมูล (Info) เช่น ช่องโหว่การใช้ MAC Algorithm ที่อ่อนแอในบริการ SSH (ตรวจพบโดย OpenVAS โดยมีค่า CVSS เท่ากับ 2.6) การไม่กำหนดค่าความมั่นคงปลอดภัยให้ cookie เช่น HttpOnly และ SameSite (ตรวจพบโดย OWASP ZAP) รวมถึงการแสดงผล HTTP Header ที่อาจนำไปใช้ในการวิเคราะห์ fingerprint ของระบบ โดยรวมแล้ว ช่องโหว่เหล่านี้จัดอยู่ในกลุ่มที่สามารถบรรเทาได้ผ่านการปรับแต่งค่าเซิร์ฟเวอร์ หรือการจำกัดการเข้าถึง เช่น การปิดพอร์ต SSH จากภายนอก การใช้ Reverse Proxy และการตั้งค่า Web Application Firewall (WAF)

ดังนั้น จึงสามารถสรุปได้ว่า ความเสี่ยงของระบบ Moodle เวอร์ชัน 4.5.1 หลังการปรับปรุงอยู่ในระดับที่ ยอมรับได้ (Acceptable Risk) ช่องโหว่ที่เหลือไม่มีผลกระทบในระดับที่เป็นอันตรายต่อความลับ ความถูกต้อง หรือความพร้อมใช้งานของข้อมูล ทั้งยังสามารถจัดการได้ภายใต้แผนการดูแลระบบที่มีประสิทธิภาพ เช่น การจัดให้มีการสแกนช่องโหว่ประจำไตรมาส การติดตามอัปเดตเวอร์ชันของปลั๊กอินและระบบหลัก รวมถึงการบังคับใช้นโยบายการควบคุมสิทธิ์ผู้ใช้งานอย่างเข้มงวด

บทที่ 5
อภิปรายผล สรุปผลการวิจัย และข้อเสนอแนะ

5.1 อภิปรายผลการวิจัย

จากผลการศึกษาชี้ชัดว่า การอัปเกรดระบบ Moodle เป็นเวอร์ชัน 4.5.1 ร่วมกับการปรับแต่งค่าความมั่นคงปลอดภัยในหลายด้าน มีประสิทธิภาพอย่างมากในการลดความเสี่ยงเชิงระบบ โดยเฉพาะช่องโหว่ในระดับ Critical และ High ซึ่งเป็นช่องโหว่ที่มีโอกาสสูงในการถูกโจมตีจากผู้ไม่หวังดี จากการตรวจสอบหลังการอัปเกรดด้วยเครื่องมือ Nessus, OWASP ZAP และ OpenVAS ไม่พบช่องโหว่ระดับรุนแรงเหล่านี้หลงเหลืออยู่เลยในเวอร์ชันใหม่ ขณะที่เวอร์ชันเก่า 3.7.1 ยังคงพบช่องโหว่ระดับสูงจำนวนมาก แม้จะมีการปรับปรุงบางส่วนแล้วก็ตาม

ผลการจัดกลุ่มช่องโหว่ (Vulnerability Grouping) ยังแสดงให้เห็นว่า ช่องโหว่ประเภท Input Validation, TLS Misconfiguration, Insecure Session และ Server Misconfiguration เป็นกลุ่มช่องโหว่ที่มีความเสี่ยงสูงและพบมากที่สุดในเวอร์ชันเดิม การมุ่งเน้นแก้ไขในกลุ่มเหล่านี้ก่อนจึงเป็นกลยุทธ์ที่มีประสิทธิภาพในการลดระดับความเสี่ยงโดยรวมของระบบ

อย่างไรก็ตาม แม้ระบบเวอร์ชันใหม่จะปลอดภัยมากขึ้น แต่ยังพบช่องโหว่ระดับต่ำ (Low) และช่องโหว่เชิงข้อมูล (Informational) บางรายการที่ไม่สามารถแก้ไขได้ทันที เช่น ช่องโหว่จากค่า Default ของโมดูลบางตัว การไม่ตั้งค่า SameSite หรือ HttpOnly ให้กับ Cookie และการใช้ MAC Algorithm ที่อ่อนแอใน SSH ซึ่งแม้จะมีผลกระทบน้อยในทางปฏิบัติ แต่ยังควรดำเนินการปรับแต่งเพิ่มเติมในระยะต่อไป

การอัปเกรดระบบเองไม่ใช่เพียงการเปลี่ยนเวอร์ชันเท่านั้น แต่เป็นกระบวนการที่ต้องวางแผนเชิงระบบ ประเมินปลั๊กอินและโมดูลที่ใช้งาน ความเข้ากันได้ของฐานข้อมูล ความพร้อมของบุคลากร และการเตรียม Staging Environment สำหรับทดสอบระบบอย่างรัดกุม โดยเฉพาะในกรณีที่ระบบเดิมมีการปรับแต่งเฉพาะทางจำนวนมาก ซึ่งอาจต้องใช้เวลาในการตรวจสอบ แก้ไข ย้ายข้อมูล และฝึกอบรมเพิ่มเติม โดยทั่วไปจะใช้ระยะเวลาเตรียมการ 3–7 วัน หรือมากกว่านั้น

นอกจากนี้ การวิเคราะห์งบประมาณยังแสดงให้เห็นว่า แม้การอัปเกรดระบบจะมีต้นทุนเริ่มต้นที่สูงกว่า แต่สามารถชดเชยค่าใช้จ่ายได้ในระยะยาว โดยมีผลประโยชน์ในด้านการลดความเสี่ยงจากเหตุการณ์ด้านความมั่นคงปลอดภัยทางไซเบอร์ และลดภาระค่าใช้จ่ายในการบำรุงรักษาระบบเดิมอย่างต่อเนื่อง ทั้งนี้ แนวทางดังกล่าวยังช่วยยกระดับความมั่นคงปลอดภัยในการให้บริการแก่ผู้ใช้งาน และส่งเสริมภาพลักษณ์ความน่าเชื่อถือของวิทยาลัยในบริบทของการศึกษายุคดิจิทัล

5.2 สรุปผลการวิจัย

การศึกษานี้มีวัตถุประสงค์เพื่อ ประเมินช่องโหว่ด้านความมั่นคงปลอดภัยของระบบจัดการเรียนการสอนออนไลน์ Moodle โดยเปรียบเทียบระหว่างเวอร์ชัน 3.7.1 และ 4.5.1 โดยใช้เครื่องมือมาตรฐาน ได้แก่ OWASP ZAP, OpenVAS และ Nessus ในการตรวจสอบช่องโหว่ทั้งก่อนและหลังการปรับปรุง จากผลการทดสอบพบว่า Moodle เวอร์ชัน 3.7.1 มีช่องโหว่จำนวนมาก โดยเฉพาะช่องโหว่ระดับ Critical และ High ที่ตรวจพบจาก Nessus รวมกันมากกว่า 60 รายการ ขณะที่ Moodle 4.5.1 ซึ่งผ่านการอัปเกรดระบบและปรับแต่งด้านความมั่นคงปลอดภัยแล้ว ไม่พบช่องโหว่ระดับ Critical และ High อีกเลย แสดงให้เห็นถึงประสิทธิภาพของการอัปเกรดระบบอย่างชัดเจน

การปรับปรุงครอบคลุมหลายมิติ เช่น การตั้งค่า SSL/TLS และ HTTP Header ที่ปลอดภัย การใช้นโยบาย Content Security Policy (CSP) การควบคุมสิทธิ์แบบ RBAC การตั้งค่าความมั่นคงปลอดภัยของ Cookie การสำรองข้อมูลอัตโนมัติ และการฝึกอบรมผู้ใช้งาน รวมถึงการทดสอบระบบและการสแกนช่องโหว่ซ้ำหลังดำเนินการแล้วเสร็จ

อย่างไรก็ตาม การอัปเกรดระบบ Moodle ไปยังเวอร์ชันใหม่ ไม่ใช่กระบวนการที่สามารถดำเนินการได้ทันทีในทุกกรณี โดยเฉพาะในองค์กรที่มีระบบเดิมขนาดใหญ่หรือมีการปรับแต่งอย่างซับซ้อน จำเป็นต้องพิจารณาปัจจัยหลายด้าน เช่น ความเข้ากันได้ของปลั๊กอินและโมดูลที่ใช้งานอยู่ ความซับซ้อนของกระบวนการอัปเกรด การปรับแต่งระบบให้รองรับโครงสร้างใหม่ และการเตรียมความพร้อมด้านการฝึกอบรมบุคลากร ทั้งหมดนี้ต้องใช้เวลาในการเตรียมการอย่างน้อย 3–7 วัน ขึ้นอยู่กับขนาดของระบบและความซับซ้อนในการเปลี่ยนแปลง

นอกจากนี้ เมื่อพิจารณาเปรียบเทียบงบประมาณระหว่างการปรับปรุงระบบเดิมกับการอัปเกรดเป็นเวอร์ชันใหม่พร้อมจัดหาเครื่องแม่ข่าย โดยอ้างอิงจากราคากลางการจ้างที่ปรึกษาของกรมบัญชีกลาง และข้อมูลราคากลางรวมถึงคุณลักษณะพื้นฐานของอุปกรณ์และระบบคอมพิวเตอร์จากกระทรวงดิจิทัล พบว่า แม้แนวทางการอัปเกรดจะมีต้นทุนเริ่มต้นที่สูงกว่า แต่สามารถให้ผลตอบแทนที่คุ้มค่าในระยะยาว เนื่องจากช่วยลดความเสี่ยงด้านความมั่นคงปลอดภัยและลดต้นทุนในการดูแลระบบในภาพรวม

5.3 ข้อเสนอแนะ

จากผลการวิจัยพบว่า การอัปเกรดระบบ Moodle เป็นเวอร์ชัน 4.5.1 ควบคู่กับการปรับปรุงด้านความมั่นคงปลอดภัยในหลายมิติ ส่งผลให้จำนวนช่องโหว่ที่ตรวจพบลดลงอย่างชัดเจน ทั้งในด้านปริมาณและระดับความรุนแรง โดยเฉพาะช่องโหว่ระดับ Critical และ High ซึ่งไม่ปรากฏอีกในเวอร์ชันใหม่หลังการอัปเกรด อย่างไรก็ตาม การอัปเกรดระบบอาจไม่สามารถดำเนินการได้ทันทีในทุกบริบท เนื่องจากต้องพิจารณาความพร้อมในหลายด้าน เช่น ความเข้ากันได้ของปลั๊กอินเดิม ทรัพยากร

ขององค์กร และระยะเวลาในการดำเนินการ ดังนั้น จึงเสนอแนวทางในการพัฒนาความมั่นคงปลอดภัยของระบบจัดการเรียนการสอนออนไลน์ให้มีประสิทธิภาพมากยิ่งขึ้น โดยสามารถแบ่งออกเป็น 5 ด้านหลัก ดังนี้

ประการแรก คือการเสริมความมั่นคงปลอดภัยของเซิร์ฟเวอร์ ควรดำเนินการปรับแต่งค่าคอนฟิกพื้นฐานของระบบให้ปลอดภัยยิ่งขึ้น เช่น การติดตั้งใบรับรอง SSL/TLS การตั้งค่า HTTP Security Header การเปิดใช้งาน Content Security Policy (CSP) และการปิดการแสดงผลข้อความแสดงข้อผิดพลาด เพื่อป้องกันการรั่วไหลของข้อมูลที่อาจนำไปสู่การโจมตีทางไซเบอร์

ประการที่สอง คือการควบคุมสิทธิ์การเข้าถึงระบบอย่างเข้มงวด โดยควรใช้หลักการ Role-Based Access Control (RBAC) ในการกำหนดระดับสิทธิ์ผู้ใช้งาน และตรวจสอบบัญชีผู้ใช้งานที่ไม่ได้ใช้เป็นประจำอย่างสม่ำเสมอ เพื่อลดโอกาสในการถูกโจมตีจากภายในระบบ หรือผ่านบัญชีที่ถูกละเลย

ประการที่สาม คือการป้องกันช่องโหว่ที่เกี่ยวข้องกับการตรวจสอบข้อมูลนำเข้า (Input Validation) โดยเฉพาะช่องโหว่ประเภท SQL Injection และ Cross-Site Scripting (XSS) ซึ่งควรใช้แนวทางการป้องกัน เช่น Parameterized Queries, Input Validation และ Output Encoding ร่วมกับการกำหนดระดับ CSP ที่เหมาะสมกับระบบ Moodle ที่ใช้งานอยู่

ประการที่สี่ คือการจัดการสแกนช่องโหว่และการติดตั้งแพตช์ด้านความมั่นคงปลอดภัยอย่างต่อเนื่อง โดยควรกำหนดรอบการสแกนเป็นรายไตรมาสด้วยเครื่องมือ เช่น OWASP ZAP, Nessus และ OpenVAS รวมถึงการตรวจสอบ Log ระบบและเหตุการณ์ผิดปกติผ่านเครื่องมือเฝ้าระวัง เพื่อให้สามารถตอบสนองต่อความเสี่ยงได้อย่างทันท่วงที แม้ว่าจะยังไม่สามารถอัปเกรด Moodle ได้ก็ตาม

ประการสุดท้าย คือการวางแผนการอัปเกรด Moodle อย่างเป็นระบบ โดยควรเตรียม Staging Environment สำหรับทดสอบก่อนการนำระบบใหม่ไปใช้งานจริง มีการสำรองข้อมูลก่อนการอัปเกรด ตรวจสอบความเข้ากันได้ของปลั๊กอิน และจัดอบรมบุคลากรล่วงหน้า ทั้งนี้ควรดำเนินการอัปเกรดในช่วงเวลาที่มีผู้ใช้งานน้อยที่สุด เช่น ช่วงปิดภาคเรียน โดยกระบวนการทั้งหมดนี้มักใช้เวลาโดยเฉลี่ยประมาณ 3–7 วัน ขึ้นอยู่กับความซับซ้อนของระบบและจำนวนปลั๊กอินที่เกี่ยวข้อง

นอกจากนี้ แนะนำให้วิทยาลัยมีนโยบายด้านความมั่นคงปลอดภัยอย่างต่อเนื่อง ทั้งในระดับเทคนิค เช่น การทำ Security Hardening และในระดับการจัดการ เช่น การกำกับดูแลการเข้าถึงข้อมูล และการจัดฝึกอบรมด้านความมั่นคงปลอดภัยอย่างสม่ำเสมอ เพื่อให้ระบบสามารถรองรับการเปลี่ยนแปลงในอนาคต และมีความมั่นคงปลอดภัยในระยะยาว หากองค์กรมีทรัพยากรเพียงพอ อาจพิจารณาบูรณาการแนวคิด DevSecOps เข้ากับกระบวนการพัฒนาระบบ เพื่อให้การอัปเดต การแก้ไข และการขยายระบบสามารถทำได้อย่างปลอดภัยในทุกขั้นตอน

บรรณานุกรม

S. Kumar, A. K. Gankotiya and K. Dutta. “A Comparative Study of Moodle with Other E-Learning Systems.” *Proceedings of the 2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, India, 2011, pp. 414–418.

“Rising Threats: 2024 Cybersecurity Trends Impacting Higher Education.” Cadre.net. (2024): https://blog.cadre.net/rising-threats-2024-cybersecurity-trends-impacting-higher-education, Accessed 6 Mar. 2024.

“Three Cybersecurity Facts Campus Leaders Should Know.” EDUCAUSE Review. (n.d.): https://er.educause.edu/articles/sponsored/2023/9/three-cybersecurity-facts-campus-leaders-should-know, Accessed 8 Sep. 2024.

A. B. Samgir, V. Gutte, K. Kolhe and D. R. Patil. “Automated Penetration Testing Architecture Using Metasploit and OWASP ZAP for Web Applications.” *Proceedings of the 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS),* Coimbatore, India, 2024, pp. 649–657.

B. S. Lakshmi, D. Kovvuri, H. N. V. Bolisetti, D. S. Chikkala, S. Karri and G. Yadlapalli. “A Proactive Approach for Detecting SQL and XSS Injection Attacks.” *Proceedings of the 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2024, pp. 1415–1420.A.

A. AlMufairej, L. BinGhaith, D. AlShareef and N. S. M. Jamail. “Cyber Security Risk Management: E-Learning System.” *Proceedings of the 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, Riyadh, Saudi Arabia, 2022, pp. 146–149.

R. Patel, U. Thakar and V. Tewari. “A Mechanism for Operation Level Role-Based Access Control in Web Services.” *Proceedings of the 2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*, Nagpur, India, 2017, pp. 384–388.

S. Ameer, J. Benson and R. Sandhu. “Hybrid Approaches (ABAC and RBAC) Toward Secure Access Control in Smart Home IoT.” *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 4032–4051.

M. A. Kunda and I. Alsmadi. “Practical Web Security Testing: Evolution of Web Application Modules and Open Source Testing Tools.” *Proceedings of the 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, San Antonio, TX, USA, 2022, pp. 152–155.

S. Tyagi and K. Kumar. “Evaluation of Static Web Vulnerability Analysis Tools.” *Proceedings of the 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Solan, India, 2018, pp. 1–6.

J. Shahid, Z. Muhammad, Z. Iqbal, M. S. Khan, Y. Amer and W. Si. “SAT: Integrated Multi-agent Blackbox Security Assessment Tool Using Machine Learning.” *Proceedings of the 2022 2nd International Conference on Artificial Intelligence (ICAI)*, Islamabad, Pakistan, 2022, pp. 105–111.

W. Qingyang, N. Yuqiao, G. Zhen, Y. Mingming, X. Shihao and C. Yang. “A Reflective XSS Vulnerability Detection Method Based on Fuzzing Test.” *Proceedings of the 2022 International Conference on Algorithms, Data Mining, and Information Technology (ADMIT)*, Xi'an, China, 2022, pp. 25–31.

C. Lv, L. Zhang, F. Zeng and J. Zhang. “Adaptive Random Testing for XSS Vulnerability.” *Proceedings of the 2019 26th Asia-Pacific Software Engineering Conference (APSEC)*, Putrajaya, Malaysia, 2019, pp. 63–69.

L. Nataly Basto Zabala, C. Santiago Rodríguez Velasco and H. Dario Jaimes Parada. "Security Scheme for Moodle Platforms Based on a Multi-layered Model." *Proceedings of the 2022 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI),* Bogota, Colombia, 2022, pp. 1–5.

S. T, J. S, B. S, J. S and A. S. Kumar. "SQL Injection Testing on Website Using Sqlmap." *Proceedings of the 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, Pune, India, 2024, pp. 1–4.

H. S. Abdullah, Z. O. Hamad and O. S. Khalind. "Analysis of SQLMAP Efficacy in Exploiting SQL Injection Vulnerabilities in Web Applications: A Case Study on DVWA." *Proceedings of the 2023 International Conference on Engineering Applied and Nano Sciences (ICEANS)*, Erbil, Iraq, 2023, pp. 13–18.

L. Min, G. Ranxin, S. Guanlin, C. Wei and X. Xiaotian. "The Detection and Defense Mechanism for SQL Injection Attack Based on Web Application." *Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 2022, pp. 1467–1470.

"Installing OpenVAS on Kali Linux." *Infotechys.* (n.d.): https://infotechys.com/installing-openvas-on-kali-linux/, Accessed 23 Feb. 2025.

กรมบัญชีกลาง. "เกณฑ์ราคากลางค่าจ้างที่ปรึกษา พ.ศ. 2567." consultant.pdmo.go.th. (2567): https://www.consultant.pdmo.go.th/docs/medium_price_new.pdf, สืบค้นเมื่อ 8 เม.ย. 2568.

กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม. "ราคากลางอุปกรณ์ไอที." mdes.go.th. (n.d.): https://www.mdes.go.th/service?a=29, สืบค้นเมื่อ 8 เม.ย. 2568.

วิทยาลัยเทคโนโลยีพณิชยการอยุธยา. "แนวทางการกำหนดอัตราค่าตอบแทนที่ปรึกษารายเดือน." เอกสารภายในวิทยาลัย, 2567. อ้างอิงโดยผู้วิจัยจากคำแนะนำของอาจารย์ที่ปรึกษา.

ภาคผนวก ก

ผลการสแกนช่องโหว่ด้วยเครื่องมือ OWASP ZAP

ตารางที่ ก-1 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก OWASP ZAP ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	SQL Injection	High	**URL :** http://110.49.6.44:8080/login/index.php **Method :** POST **Attack :** Hjpk3l3Gs1OpqUZU6DJ6g6rr0VH8GN0N AND 1=1 -- **Other Info :** The page results were successfully manipulated using the boolean conditions [Hjpk3l3Gs1OpqUZU6DJ6g6rr0VH8GN0N AND 1=1 --] and [Hjpk3 l3 Gs1 OpqUZU6 DJ6 g6 rr0 VH8 GN0 N AND 1 = 2 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.
2.	SQL Injection	High	**URL :** http://110.49.6.44:8080/login/index.php **Method :** POST **Attack :** x4Wqbd8w67qJqef3ZFXgDbQPHwlGKJJU OR 1=1 -- **Other Info :** The page results were successfully manipulated using the boolean conditions [x4Wqbd8w67qJqef3ZFXgDbQPHwlGKJJU AND 1=1 --] and [x4Wqbd8w67qJqef3ZFXgDbQPHwlGKJJU OR 1=1 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter.
3.	SQL Injection	High	**URL :** http://110.49.6.44:8080/login/index.php **Method :** POST **Attack :** 2/2 **Other Info :** The original page results were successfully replicated using the expression [2/2] as the parameter value The parameter value being modified was stripped from the HTML output for the purposes of the comparison.
4.	SQL Injection	High	**URL :** http://110.49.6.44:8080/login/index.php **Method :** POST **Attack :** guest' AND '1'='1' -- **Other Info :** The page results were successfully manipulated using the boolean conditions [guest' AND '1'='1' --] and [guest' AND '1'='2' --] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.

ตารางที่ ก-1 (ต่อ)

No.	Name	Level	Descriptiom
5.	Content Security Policy (CSP) Header Not Set	Medium	**URL :** http://110.49.6.44:8080 **URL :** http://110.49.6.44:8080/?lang=en **URL :** http://110.49.6.44:8080/?lang=th **URL :** http://110.49.6.44:8080/admin/tool/dataprivacy/summary.php **URL :** http://110.49.6.44:8080/admin/tool/dataprivacy/summary.php?lang=en **URL :** http://110.49.6.44:8080/admin/tool/dataprivacy/summary.php?lang=th **URL :** http://110.49.6.44:8080/login/forgot_password.php **URL :** http://110.49.6.44:8080/login/forgot_password.php?lang=en **URL :** http://110.49.6.44:8080/login/forgot_password.php?lang=th **URL :** http://110.49.6.44:8080/login/index.php **URL :** http://110.49.6.44:8080/robots.txt **URL :** http://110.49.6.44:8080/sitemap.xml **Method :** GET **Description :** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
6.	Hidden File Found	Medium	**URL :** http://110.49.6.44:8080/composer.lock **Method :** GET **Evidence :** HTTP/1.1 200 OK **Other Info :** composer
7.	Vulnerable JS Library	Medium	**URL :** http://110.49.6.44:8080/lib/javascript.php/1735910940/lib/jquery/jquery-3.2.1.min.js **Method :** GET **Evidence :** jquery-3.2.1.min.js **Other Info :** CVE-2020-11023 CVE-2020-11022 CVE-2019-11358

จากการสแกน Moodle 3.7.1 ก่อนการแก้ไข พบช่องโหว่ด้านความมั่นคงปลอดภัย รวม 7 รายการ ตามที่แสดงใน ตารางที่ ก-1 โดยแบ่งเป็น ช่องโหว่ระดับ High 4 รายการ และ Medium 3 รายการ ช่องโหว่ระดับ High ทั้งหมดเป็น SQL Injection ซึ่งตรวจพบที่หน้า login/index.php และอาจถูกใช้เพื่อเข้าถึงหรือแก้ไขข้อมูลในฐานข้อมูลโดยไม่ได้รับอนุญาต นอกจากนี้ ช่องโหว่ระดับ Medium ประกอบด้วย Content Security Policy (CSP) Header Not Set ซึ่งเพิ่มความเสี่ยงต่อ

Cross-Site Scripting (XSS) และ Data Injection Attack, Hidden File Found ซึ่งอาจเปิดเผยไฟล์ที่มีข้อมูลสำคัญ และ Vulnerable JavaScript Library ที่ใช้ jQuery เวอร์ชันเก่าที่มีช่องโหว่ด้านความมั่นคงปลอดภัย

เพื่อป้องกันความเสี่ยง ควร ปิดกั้น SQL Injection โดยใช้ Parameterized Queries หรือ Prepared Statements กำหนดค่า Content Security Policy (CSP) Header เพื่อลดความเสี่ยงจาก XSS และ ClickJacking ลบไฟล์ที่ไม่จำเป็นที่อาจรั่วไหลข้อมูลสำคัญ และอัปเดตไลบรารี JavaScript ให้เป็นเวอร์ชันล่าสุด พร้อมทั้ง ดำเนินการ re-scan หลังการแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขแล้ว

ตารางที่ ก-2 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก OWASP ZAP ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	SQL Injection	High	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Attack :** sGE6T2GohZz2EV6EKqjWfamDiHtj2ryR OR 1=1 -- **Other Info :** The page results were successfully manipulated using the boolean conditions [sGE6T2GohZz2EV6EKqjWfamDiHtj2ryR AND 1=1 --] and [sGE6T2GohZz2EV6EKqjWfamDiHtj2ryR OR 1=1 --] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison. Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter.
2.	SQL Injection	High	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Attack :** VKHJTgkTsQa59bgAcclMhnTn5N6Bcgum OR 1=1 -- **Other Info :** The page results were successfully manipulated using the boolean conditions [VKHJTgkTsQa59bgAcclMhnTn5N6Bcgum AND 1=1 --] and [VKHJTgkTsQa59bgAcclMhnTn5N6Bcgum OR 1=1 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was NOT returned for the original parameter. The vulnerability was detected by successfully retrieving more data than originally returned, by manipulating the parameter.
3.	SQL Injection	High	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Attack :** ZAP AND 1=1 - **Other Info :** The page results were successfully manipulated using the boolean conditions [ZAP AND 1=1 --] and [ZAP AND 1=2 --] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.
4.	SQL Injection	High	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Attack :** ZAP AND 1=1 - **Other Info :** The page results were successfully manipulated using the boolean conditions [ZAP' AND '1'='1' --] and [ZAP' AND '1'='2' --] The parameter value being modified was NOT stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.

ตารางที่ ก-2 (ต่อ)

No.	Name	Level	Descriptiom
5.	Absence of Anti-CSRF Tokens	Medium	**URL :** http://110.49.6.45:8080/login/forgot_password.php **Method :** POST **Evidence :** <form autocomplete="off" action="http://110.49.6.45:8080/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_FhPO20APFPbryHE" class="mform"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "_qf__login_forgot_password_form" "id_email" "id_submitbuttonemail" "id_submitbuttonusername" "id_username" "sesskey"].
6.	Absence of Anti-CSRF Tokens	Medium	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Evidence :** <form class="login-form" action="http://110.49.6.45:8080/login/index.php" method="post" id="login"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "anchor" "logintoken" "password" "username"].
7.	Absence of Anti-CSRF Tokens	Medium	**URL :** http://110.49.6.45:8080/login/index.php **Method :** POST **Evidence :** <form action="http://110.49.6.45:8080/login/index.php" method="post" id="guestlogin"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "anchor" "logintoken" "password" "username"].

ตารางที่ ก-2 (ต่อ)

No.	Name	Level	Descriptiom
8.	Content Security Policy (CSP) Header Not Set	Medium	**URL :** http://110.49.6.45:8080 **URL :** http://110.49.6.45:8080/?lang=en **URL :** http://110.49.6.45:8080/?lang=th **URL :** http://110.49.6.45:8080/admin/tool/dataprivacy/summary.php **URL :** http://110.49.6.45:8080/admin/tool/dataprivacy/summary.php?lang=en **URL :** http://110.49.6.45:8080/admin/tool/dataprivacy/summary.php?lang=th **URL :** http://110.49.6.45:8080/login/forgot_password.php **URL :** http://110.49.6.45:8080/login/index.php **URL :** http://110.49.6.45:8080/login/index.php?lang=en **URL :** http://110.49.6.45:8080/login/index.php?lang=th **URL :** http://110.49.6.45:8080/login/index.php?loginredirect=1 **URL :** http://110.49.6.45:8080/robots.txt **URL :** http://110.49.6.45:8080/sitemap.xml **Method :** GET **Description :** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

จากการสแกน Moodle 4.5.1 ก่อนการแก้ไข พบช่องโหว่ด้านความมั่นคงปลอดภัย รวม 8 รายการ ตามที่แสดงใน ตารางที่ ก-2 โดยแบ่งเป็น ช่องโหว่ระดับ High 4 รายการ และ Medium 4 รายการ ช่องโหว่ระดับ High ทั้งหมดเป็น SQL Injection ซึ่งพบที่หน้า login/index.php และสามารถถูกใช้เพื่อเข้าถึงหรือแก้ไขข้อมูลในฐานข้อมูลโดยไม่ได้รับอนุญาต นอกจากนี้ ช่องโหว่ระดับ Medium ประกอบด้วย Absence of Anti-CSRF Tokens ซึ่งทำให้ระบบเสี่ยงต่อ Cross-Site Request Forgery (CSRF) Attack และ Content Security Policy (CSP) Header Not Set ซึ่งอาจเปิดโอกาสให้เกิดการโจมตีแบบ Cross-Site Scripting (XSS) และ Data Injection

เพื่อป้องกันความเสี่ยง ควร ปิดกั้น SQL Injection โดยใช้ Parameterized Queries หรือ Prepared Statements เพิ่ม CSRF Tokens ในฟอร์มสำคัญทั้งหมด และกำหนด Content Security Policy (CSP) Header เพื่อลดความเสี่ยงจาก XSS และ ClickJacking พร้อมทั้ง ดำเนินการ re-scan หลังการแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขแล้ว

ตารางที่ ก-3 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก OWASP ZAP หลังการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	SQL Injection	High	**URL :** https://olm371test.duckdns.org/lib/ajax/service.php?sesskey=Dt3REojiuL&info=core_calendar_get_calendar_monthly_view **Method :** POST **Attack :** 0 AND 1=1 -- **Other Info :** The page results were successfully manipulated using the boolean conditions [0 AND 1=1 --] and [0 AND 1=2 --] The parameter value being modified was stripped from the HTML output for the purposes of the comparison. Data was returned for the original parameter. The vulnerability was detected by successfully restricting the data originally returned, by manipulating the parameter.
2.	Absence of Anti-CSRF Tokens	Medium	**URL :** https://olm371test.duckdns.org/course/recent.php **Method :** POST **Evidence :** <form autocomplete="off" action="https://olm371test.duckdns.org/course/recent.php" method="post" accept-charset="utf-8" id="mform1_3hg7mZXYsPGbL1C" class="mform"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "_qf__recent_form" "id" "id_date_enabled" "id_submitbutton" "mform_isexpanded_id_filters" "mform_showmore_id_filters" "sesskey" "user"].

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Description
3.	Absence of Anti-CSRF Tokens	Medium	**URL :** https://olm371test.duckdns.org/login/forgot_password.php **Method :** POST **Evidence :** - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_9aaMcotbJ8YVt1t" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_AFksvZ0pkjGHyga" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_GCd7I3KQw6t7qQS" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_Ihq1hWgOpnWlg5A" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_KjA7teSXoTL9typ" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_MIZBseSN8GHJ45q" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_X20GCJtCwo09E5f" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_Yk3BM8NIjgGtuaD" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_fSK5CU2VHOG9rrT" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_irhq2gAWhfWrZuc" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_oLbLpqvDdrXWv5M" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_pbSrdLpQxoA2DSD" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_s4qPTYBEh3zYHG7" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_s5ysF1QxhZBg3jV" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_vMVggfN5ocvWULo" class="mform"> - <form autocomplete="off" action="https://olm371test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_wPWYO4YXvYtFqEE" class="mform"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "_qf__login_forgot_password_form" "id_email" "id_submitbuttonemail" "id_submitbuttonusername" "id_username" "sesskey"].

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Descriptiom
4.	Absence of Anti-CSRF Tokens	Medium	**URL :** https://olm371test.duckdns.org/login/index.php **Method :** POST **Evidence :** - <form class="mt-3" action="https://olm371test.duckdns.org/login/index.php" method="post" id="login"> - <form action="https://olm371test.duckdns.org/login/index.php" method="post" id=" guestlogin"> **Other Info :** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 1: "anchor" "logintoken" "password" "rememberusername" "username"].
5.	Content Security Policy (CSP) Header Not Set	Medium	**URL :** https://olm371test.duckdns.org **URL :** https://olm371test.duckdns.org/?lang=en **URL :** https://olm371test.duckdns.org/?lang=th **URL :** https://olm371test.duckdns.org/admin/tool/dataprivacy/summary.php **URL :** https://olm371test.duckdns.org/admin/tool/dataprivacy/summary.php?lang=en **URL :** https://olm371test.duckdns.org/admin/tool/dataprivacy/summary.php?lang=th **URL :** https://olm371test.duckdns.org/calendar/export.php **URL :** https://olm371test.duckdns.org/calendar/view.php?view=day&time=1740009600 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month&time=1727740800 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month&time=1735689600 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month&time=1738368000 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month&time=1740009600&lang=en **URL :** https://olm371test.duckdns.org/calendar/view.php?view=month&time=1740009600&lang=th **URL :** https://olm371test.duckdns.org/calendar/view.php?view=upcoming&course=1&time=1740066879

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Descriptiom
			URL : https://olm371test.duckdns.org/calendar/view.php?view=upcoming&course=1&time=1740066971 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=upcoming&course=1&time=1740066976 **URL :** https://olm371test.duckdns.org/calendar/view.php?view=upcoming&course=1&time=1740067058 **URL :** https://olm371test.duckdns.org/course/recent.php?id=1 **URL :** https://olm371test.duckdns.org/course/recent.php?id=1&lang=en **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=axP4raDEFr&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=axP4raDEFr&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=CrIA9aeU6E&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=cVZP87pCZD&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=cVZP87pCZD&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=eGAZb9H68z&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=eGAZb9H68z&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ewTvsumUMN&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ewTvsumUMN&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=HfczLyq8d0&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=HfczLyq8d0&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=j564i25kpW&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ocACvGxywV&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ocACvGxywV&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=pwSHV3d3aN&pref=drawer-open-nav&value=false

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Descriptiom
			URL : https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=pwSHV3d3aN&pref=drawer-open-nav&value=true **Method :** GET **Description :** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
6.	Hidden File Found	Medium	**URL :** https://olm371test.duckdns.org/composer.lock **Method :** GET **Evidence :** HTTP/1.1 200 OK **Description:** A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.
7.	Missing Anti-clickjacking Header	Medium	**URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=axP4raDEFr&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=axP4raDEFr&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=CrIA9aeU6E&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=cVZP87pCZD&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=cVZP87pCZD&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=eGAZb9H68z&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=eGAZb9H68z&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ewTvsumUMN&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ewTvsumUMN&pref=drawer-open-nav&value=true

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Description
			URL : https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=HfczLyq8d0&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=HfczLyq8d0&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=j564i25kpW&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ocACvGxywV&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ocACvGxywV&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=pwSHV3d3aN&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=pwSHV3d3aN&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=Rkf5ksUomB&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=Rkf5ksUomB&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=TMTgFQxMRp&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ve8PkZdue4&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ve8PkZdue4&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=vLGxCl9L8T&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=vLGxCl9L8T&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=x36dPhr9yM&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=xgs8YMw5Cb&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=xgs8YMw5Cb&pref=drawer-open-nav&value=true **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=xWUSJG6Zpm&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=xWUSJG6Zpm&pref=drawer-open-nav&value=true

ตารางที่ ก-3 (ต่อ)

No.	Name	Level	Descriptiom
			URL : https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ZLwzHufKWJ&pref=drawer-open-nav&value=false **URL :** https://olm371test.duckdns.org/lib/ajax/setuserpref.php?sesskey=ZLwzHufKWJ&pref=drawer-open-nav&value=true **Method :** GET **Evidence :** HTTP/1.1 200 OK **Description:** The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
8.	Vulnerable JS Library	Medium	**URL :** https://olm371test.duckdns.org/lib/javascript.php/1735910940/lib/jquery/jquery-3.2.1.min.js **Method :** GET **Evidence :** jquery-3.2.1.min.js **Other Info:** CVE-2020-11023 CVE-2020-11022 CVE-2019-11358.

หลังจากการแก้ไขช่องโหว่ของ Moodle 3.7.1 ระบบยังคงมีช่องโหว่ด้านความมั่นคงปลอดภัยอยู่ 8 รายการ ตามที่แสดงใน ตารางที่ ก-3 โดยประกอบด้วยช่องโหว่ระดับ High 1 รายการ และ Medium 7 รายการ ช่องโหว่ที่ยังพบ ได้แก่ SQL Injection, Absence of Anti-CSRF Tokens, Content Security Policy (CSP) Header Not Set, Hidden File Found, Missing Anti-Clickjacking Header และ Vulnerable JavaScript Library ช่องโหว่เหล่านี้อาจทำให้ระบบเสี่ยงต่อการโจมตีแบบ SQL Injection, Cross-Site Request Forgery (CSRF), ClickJacking และ Cross-Site Scripting (XSS) รวมถึงการใช้ไลบรารี JavaScript เวอร์ชันเก่าที่มีช่องโหว่

เพื่อป้องกันความเสี่ยงเพิ่มเติม ควร ป้องกัน SQL Injection โดยใช้ Prepared Statements หรือ Parameterized Queries เพิ่ม CSRF Tokens ในแบบฟอร์มที่สำคัญ กำหนด Content Security Policy (CSP) และ X-Frame-Options เพื่อลดความเสี่ยงจาก XSS และ ClickJacking, ลบไฟล์ที่ไม่จำเป็นที่อาจรั่วไหลข้อมูลสำคัญ และอัปเดตไลบรารี JavaScript ให้เป็นเวอร์ชันล่าสุด พร้อมทั้ง ดำเนินการ re-scan หลังการแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขแล้ว

ตารางที่ ก-4 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก OWASP ZAP หลังการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	Absence of Anti-CSRF Tokens	Medium	**URL :** https://olm451test.duckdns.org/login/forgot_password.php **Method :** POST **Evidence :** <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_6XBKN5JBaaycPdj" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_BEzaFMOXgpob3p8" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_BIktK0mN19oKF8P" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_BWppiW4zIUKduY9" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_BlLmHQO6DaDDYnZ" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_C8jxSYqdnwIBH68" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_I7lqFQCAhWwhs1c" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_O8FAGXUMwUExBfG" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_Q4iRk6qwBFKrB45" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_RGBThgBGvHfuFy4" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_Rq5yBz1P4gbDyuv" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_SNpfGsxQgckMHCZ" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_SuNUOESNCAiDlaq" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_VnwgpGv3PcFQHte" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_YltCHzcNROzlySQ" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_cBEPuGfax6efP8r" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_d9u6gYJ0ICrlyDZ" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_dvCUDeGGPqLBa5H" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_odHDpAE2flGOCiN" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_os6bufKslvWHP5Y" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_qpFASOtobtwjyCD" class="mform"> <form autocomplete="off" action="https://olm451test.duckdns.org/login/forgot_password.php" method="post" accept-charset="utf-8" id="mform1_yyM2rvKuA8Z9KLw" class="mform"> <form class="login-form" action="https://olm451test.duckdns.org/login/index.php" method="post" id="login">

ตารางที่ ก-4 (ต่อ)

No.	Name	Level	Descriptiom
			<form action="https://olm451test.duckdns.org/login/index.php" method="post" id="guestlogin"> **Other Info:** No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token] was found in the following HTML form: [Form 2: "logintoken" "password" "username"].
2.	Content Security Policy (CSP) Header Not Set	Medium	**URL :** https://olm451test.duckdns.org/ **URL :** https://olm451test.duckdns.org/?lang=en **URL :** https://olm451test.duckdns.org/?lang=th **URL :** https://olm451test.duckdns.org/admin/tool/dataprivacy/summary.php **URL :** https://olm451test.duckdns.org/admin/tool/dataprivacy/summary.php?lang=en **URL :** https://olm451test.duckdns.org/admin/tool/dataprivacy/summary.php?lang=th **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041522&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041532&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041551&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041559&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041569&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041584&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041588&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041592&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041598&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041607&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041620&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041626&course=1

ตารางที่ ก-4 (ต่อ)

No.	Name	Level	Descriptiom
			URL : https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041638&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041641&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041647&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041650&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041669&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041678&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041685&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041686&course=1 **URL :** https://olm451test.duckdns.org/calendar/view.php?view=month&time=1740041690&course=1 **Method :** GET **Description :** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

จากการสแกน Moodle 4.5.1 หลังการแก้ไข พบว่ายังคงมีช่องโหว่ระดับ Medium จำนวน 2 รายการ ตามที่แสดงใน ตารางที่ ก-4 ได้แก่ Absence of Anti-CSRF Tokens และ Content Security Policy (CSP) Header Not Set โดยช่องโหว่แรกเกิดจากการที่บางแบบฟอร์มในระบบไม่มีการใช้ CSRF Token ซึ่งอาจทำให้เกิด Cross-Site Request Forgery (CSRF) Attack ส่งผลให้ผู้ไม่หวังดีสามารถส่งคำขอปลอมแทนผู้ใช้ที่ล็อกอินอยู่ได้ ส่วนช่องโหว่ที่สองเกิดจากการที่ระบบไม่ได้กำหนด CSP Header ทำให้เสี่ยงต่อ Cross-Site Scripting (XSS) และ Data Injection Attack ซึ่งอาจนำไปสู่การขโมยข้อมูลหรือเปลี่ยนแปลงเนื้อหาของเว็บไซต์ เพื่อป้องกันความเสี่ยงเหล่านี้ ควร เพิ่ม CSRF Token ในทุกฟอร์ม

ที่สำคัญ และกำหนด Content Security Policy (CSP) Header ให้เหมาะสม พร้อมทั้ง ดำเนินการ re-scan หลังจากแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขเรียบร้อยแล้ว

ภาคผนวก ข

ผลการสแกนช่องโหว่ด้วยเครื่องมือ OpenVAS

ตารางที่ ข-1 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก OpenVAS ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	SSL/TLS: Certi cate Expired	Medium	Product Detection Result Product: cpe:/a:ietf:transport_layer_security Method: SSL/TLS: Collect and Report Certificate Details OID: 1.3.6.1.4.1.25623.1.0.103692)
2.	Missing 'HttpOnly' Cookie Attribute (HTTP)	Medium	The remote web server/application does not set the 'HttpOnly' attribute for one or more session cookies, making them accessible via JavaScript, which could lead to session hijacking attacks.
3.	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 ProtocolDetection	Medium	Product detection result cpe:/a:ietf:transport_layer_security:1.0 Detected by SSL/TLS: Version Detection (OID: 1.3.6.1.4.1.25623.1.0.105782)
4.	SSL/TLS: Certi cate Signed Using A Weak Signature Algorithm	Medium	The remote service is using a SSL/TLS certicate in the certicate chain that has been signed using a cryptographically weak hashing algorithm.
5.	Missing 'HttpOnly' Cookie Attribute (HTTP)	Medium	The remote HTTP web server / application is missing to set the 'HttpOnly' cookie attribute for one or more sent HTTP cookie.
6.	Cleartext Transmissionof Sensitive Information via HTTP	Medium	Thehost / application transmits sensitive information (username, passwords) in cleartext via HTTP.

จากการสแกน Moodle 3.7.1 ก่อนการแก้ไข พบช่องโหว่ด้านความมั่นคงปลอดภัย 6 รายการ ซึ่งอยู่ในระดับ Medium ตามที่แสดงใน ตารางที่ ข-1 โดยช่องโหว่ที่พบหลัก ๆ ได้แก่ ใบรับรอง SSL หมดอายุ (Expired SSL Certificate) ใบรับรอง SSL ที่ใช้ลายเซ็นที่ไม่แข็งแกร่ง (Weak Signature Algorithm) การรองรับโปรโตคอล TLS 1.0 และ 1.1 ที่ล้าสมัย การส่งข้อมูลสำคัญผ่าน HTTP แบบไม่เข้ารหัส (Cleartext Transmission of Sensitive Information) และการไม่กำหนดค่า 'HttpOnly' บนคุกกี้ (HttpOnly Cookie Attribute Missing)

ช่องโหว่เหล่านี้อาจทำให้ระบบเสี่ยงต่อ Man-in-the-Middle (MitM) Attack, Session Hijacking และการดักฟังข้อมูลที่ส่งผ่านเครือข่าย เพื่อป้องกันความเสี่ยง ควร ติดตั้งและใช้ใบรับรอง SSL ที่น่าเชื่อถือ ปิดการรองรับ TLS 1.0 และ 1.1 แล้วใช้ TLS 1.2 หรือ 1.3 แทน กำหนดค่า

'HttpOnly' บนคุกกี้เพื่อป้องกันการโจมตีผ่าน JavaScript และบังคับใช้ HTTPS สำหรับทุกการเชื่อมต่อ พร้อมทั้งดำเนินการ re-scan หลังจากแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขเรียบร้อยแล้ว

ตารางที่ ข-2 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก OpenVAS ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	Cleartext Transmission of Sensitive Information via HTTP	Medium	The host / application transmits sensitive information (username, passwords) in cleartext via HTTP.

จากการสแกน Moodle 4.5.1 ก่อนการแก้ไข พบช่องโหว่ Cleartext Transmission of Sensitive Information via HTTP ซึ่งอยู่ในระดับ Medium ตามที่แสดงใน ตารางที่ ข-2 ช่องโหว่นี้เกิดจากการที่ระบบส่งข้อมูลที่ละเอียดอ่อน เช่น ชื่อผู้ใช้และรหัสผ่านผ่าน HTTP แบบไม่เข้ารหัส ซึ่งอาจทำให้ผู้ไม่หวังดีดักจับข้อมูลและนำไปใช้ในการโจมตี เช่น Man-in-the-Middle (MitM) Attack

เพื่อแก้ไขปัญหานี้ ควร บังคับใช้ HTTPS แทน HTTP โดยติดตั้งใบรับรอง SSL ที่น่าเชื่อถือและกำหนดให้เว็บเซิร์ฟเวอร์บังคับใช้งาน HTTPS สำหรับทุกการเชื่อมต่อ นอกจากนี้ ควรตั้งค่า HTTP Strict Transport Security (HSTS) เพื่อป้องกันการใช้งานโปรโตคอลที่ไม่ปลอดภัย และตรวจสอบให้แน่ใจว่าข้อมูลรับ-ส่งทั้งหมดมีการเข้ารหัสที่เหมาะสม เพื่อปกป้องความเป็นส่วนตัวของผู้ใช้งานและลดความเสี่ยงด้านความมั่นคงปลอดภัย

ตารางที่ ข-3 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก OpenVAS หลังการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
1.	Missing 'HttpOnly' Cookie Attribute (HTTP)	Medium	The remote HTTP web server / application is missing to set the 'HttpOnly' cookie attribute for one or more sent HTTP cookie.
2.	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	Medium	Productdetectionresult cpe:/a:ietf:transport_layer_security:1.0 Detected by SSL/TLS: Version Detection (OID:1.3.6.1.4.1.25623.1.0.105782)

หลังจากดำเนินการแก้ไขช่องโหว่ของ Moodle 3.7.1 แล้ว OpenVAS พบช่องโหว่ที่เหลืออยู่เพียง 2 รายการในระดับ Medium ได้แก่ Missing 'HttpOnly' Cookie Attribute และ การใช้โปรโตคอล TLS 1.0 และ TLS 1.1 ที่ล้าสมัย ตามที่แสดงใน ตารางที่ ข-3 ช่องโหว่แรกอาจทำให้เกิดความเสี่ยงจาก Session Hijacking จึงควรตั้งค่า 'HttpOnly' บนคุกกี้ที่เกี่ยวข้องเพื่อลดความเสี่ยง ส่วนช่องโหว่ที่สองชี้ให้เห็นว่าระบบยังรองรับโปรโตคอลเข้ารหัสที่ไม่ปลอดภัย ซึ่งควรปิดใช้งาน TLS 1.0 และ 1.1 แล้วใช้ TLS 1.2 หรือ 1.3 แทน แม้ว่าช่องโหว่ระดับ Critical และ High จะถูกกำจัดไปแล้ว แต่ยังต้องดำเนินการแก้ไขเพิ่มเติมเพื่อเพิ่มความมั่นคงปลอดภัยของระบบให้สมบูรณ์ยิ่งขึ้น

ตารางที่ ข-4 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก OpenVAS หลังการแก้ไขช่องโหว่

No.	Name	Level	Descriptiom
-	-	-	-
-	-	-	-

หลังการสแกน Moodle 4.5.1 หลังการแก้ไข พบว่า ไม่มีช่องโหว่เหลืออยู่ ตามที่แสดงในตารางที่ ข-4 ซึ่งแสดงให้เห็นว่ามาตรการแก้ไขที่ดำเนินการไปมีประสิทธิภาพในการลดความเสี่ยงด้านความมั่นคงปลอดภัย อย่างไรก็ตาม ควรดำเนินการ ตรวจสอบและติดตามการอัปเดตความมั่นคงปลอดภัยของระบบอย่างต่อเนื่อง รวมถึง ทำการสแกนเป็นระยะ เพื่อให้มั่นใจว่าระบบยังคงปลอดภัยจากช่องโหว่ที่อาจเกิดขึ้นในอนาคต

ภาคผนวก ค

ผลการสแกนช่องโหว่ด้วยเครื่องมือ Nessus

ตารางที่ ค-1 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก Nessus ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descrptiom
1.	139574 - Apache 2.4.x < 2.4.46 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.46. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.46 advisory. - Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE (CVE-2020-11984) - Apache HTTP Server versions 2.4.20 to 2.4.43 When trace/debug was enabled for the HTTP/2 module and on certain traffic edge patterns, logging statements were made on the wrong connection, causing concurrent use of memory pools. Configuring the LogLevel of mod_http2 above info will mitigate this vulnerability for unpatched servers. (CVE-2020-11993) - Apache HTTP Server versions 2.4.20 to 2.4.43. A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards. Configuring the HTTP/2 feature via H2Push off will mitigate this vulnerability for unpatched servers. (CVE-2020-9490) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
2.	150280 - Apache 2.4.x < 2.4.47 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.47. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.47 changelog: - Unexpected <Location> section matching with 'MergeSlashes OFF' (CVE-2021-30641) - mod_auth_digest: possible stack overflow by one nul byte while validating the Digest nonce. (CVE-2020-35452) - mod_session: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service with a malicious backend server and SessionHeader. (CVE-2021-26691) - mod_session: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service. (CVE-2021-26690) - mod_proxy_http: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service. (CVE-2020-13950) - Windows: Prevent local users from stopping the httpd process (CVE-2020-13938) - mod_proxy_wstunnel, mod_proxy_http: Handle Upgradable protocols end-to-end negotiation. (CVE-2019-17567) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
3.	161454 - Apache 2.4.x < 2.4.52 mod_lua Buffer Overflow	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.52. It is, therefore, affected by a flaw related to mod_lua when handling multipart content. A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser (r:parsebody() called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability though it might be possible to craft one. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
4.	158900 - Apache 2.4.x < 2.4.53 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.53. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.53 advisory. - mod_lua Use of uninitialized value of in r:parsebody: A carefully crafted request body can cause a read to a random memory area which could cause the process to crash. This issue affects Apache HTTP Server 2.4.52 and earlier. Acknowledgements: Chamal De Silva (CVE-2022-22719) - HTTP request smuggling: Apache HTTP Server 2.4.52 and earlier fails to close inbound connection when errors are encountered discarding the request body, exposing the server to HTTP Request Smuggling Acknowledgements: James Kettle <james.kettle portswigger.net> (CVE-2022-22720) - Possible buffer overflow with very large or unlimited LimitXMLRequestBody in core: If LimitXMLRequestBody is set to allow request bodies larger than 350MB (defaults to 1M) on 32 bit systems an integer overflow happens which later causes out of bounds writes. This issue affects Apache HTTP Server 2.4.52 and earlier. Acknowledgements: Anonymous working with Trend Micro Zero Day Initiative (CVE-2022-22721) - Read/write beyond bounds in mod_sed: Out-of-bounds Write vulnerability in mod_sed of Apache HTTP Server allows an attacker to overwrite heap memory with possibly attacker provided data. This issue affects Apache HTTP Server 2.4 version 2.4.52 and prior versions. Acknowledgements: Ronald Crane (Zippenhop LLC) (CVE-2022-23943) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
5.	193421 - Apache 2.4.x < 2.4.54 Authentication Bypass	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by an authentication bypass vulnerability as referenced in the 2.4.54 advisory. - X-Forwarded-For dropped by hop-by-hop mechanism in mod_proxy: Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side Connection header hop-by-hop mechanism. This may be used to bypass IP based authentication on the origin server/application. Acknowledgements: The Apache HTTP Server project would like to thank Gaetan Ferry (Synacktiv) for reporting this issue Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
6.	161948 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Read beyond bounds via ap_rwrite(): The ap_rwrite() function in Apache HTTP Server 2.4.53 and earlier may read unintended memory if an attacker can cause the server to reflect very large input using ap_rwrite() or ap_rputs(), such as with mod_luas r:puts() function. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28614) - Read beyond bounds in ap_strcmp_match(): Apache HTTP Server 2.4.53 and earlier may crash or disclose information due to a read beyond bounds in ap_strcmp_match() when provided with an extremely large input buffer. While no code distributed with the server can be coerced into such a call, third-party modules or lua scripts that use ap_strcmp_match() may hypothetically be affected. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28615) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
7.	170113 - Apache 2.4.x < 2.4.55 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.55. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.55 advisory. - A carefully crafted If: request header can cause a memory read, or write of a single zero byte, in a pool (heap) memory location beyond the header value sent. This could cause the process to crash. This issue affects Apache HTTP Server 2.4.54 and earlier. (CVE-2006-20001) - Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.54 and prior versions. (CVE-2022-36760) - Prior to Apache HTTP Server 2.4.55, a malicious backend can cause the response headers to be truncated early, resulting in some headers being incorporated into the response body. If the later headers have any security purpose, they will not be interpreted by the client. (CVE-2022-37436) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
8.	172186 - Apache 2.4.x < 2.4.56 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.56. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.56 advisory. - HTTP request splitting with mod_rewrite and mod_proxy: Some mod_proxy configurations on Apache HTTP Server versions 2.4.0 through 2.4.55 allow a HTTP Request Smuggling attack. Configurations are affected when mod_proxy is enabled along with some form of RewriteRule or ProxyPassMatch in which a non-specific pattern matches some portion of the user-supplied request-target (URL) data and is then re-inserted into the proxied request-target using variable substitution. For example, something like: RewriteEngine on RewriteRule ^/here/(.*) http://example.com:8080/elsewhere?$1 http://example.com:8080/elsewhere ; [P] ProxyPassReverse /here/ http://example.com:8080/ http:// example.com:8080/ Request splitting/smuggling could result in bypass of access controls in the proxy server, proxying unintended URLs to existing origin servers, and cache poisoning. Acknowledgements: finder: Lars Krapf of Adobe (CVE-2023-25690) - Apache HTTP Server: mod_proxy_uwsgi HTTP response splitting: HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client. Acknowledgements: finder: Dimas Fariski Setyawan Putra (nyxsorcerer) (CVE-2023-27522) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
9.	201198 - Apache 2.4.x < 2.4.60 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.60. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.60 advisory. - Serving WebSocket protocol upgrades over a HTTP/2 connection could result in a Null Pointer dereference, leading to a crash of the server process, degrading performance. (CVE-2024-36387) - SSRF in Apache HTTP Server on Windows allows to potentially leak NTML hashes to a malicious server via SSRF and malicious requests or content Users are recommended to upgrade to version 2.4.60 which fixes this issue. Note: Existing configurations that access UNC paths will have to configure new directive UNCList to allow access during request processing. (CVE-2024-38472) - Encoding problem in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows request URLs with incorrect encoding to be sent to backend services, potentially bypassing authentication via crafted requests. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38473) - Substitution encoding issue in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows attacker to execute scripts in directories permitted by the configuration but not directly reachable by any URL or source disclosure of scripts meant to only to be executed as CGI. Users are recommended to upgrade to version 2.4.60, which fixes this issue. Some RewriteRules that capture and substitute unsafely will now fail unless rewrite flag UnsafeAllow3F is specified. (CVE-2024-38474)- Improper escaping of output in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to map URLs to filesystem locations that are permitted to be served by the server but are not intentionally/ directly reachable by any URL, resulting in code execution or source code disclosure. Substitutions in server context that use a backreferences or variables as the first segment of the substitution are affected. Some unsafe RewiteRules will be broken by this change and the rewrite flag UnsafePrefixStat can be used to opt back in once ensuring the substitution is appropriately constrained. (CVE-2024-38475) - Vulnerability in core of Apache HTTP Server 2.4.59 and earlier are vulnerably to information disclosure, SSRF or local script execution via backend applications whose response headers are malicious or exploitable. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38476) - null pointer dereference in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the server via a malicious request. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38477) - Potential SSRF in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe RewriteRules to unexpectedly setup URL's to be handled by mod_proxy. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-39573) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
10.	156255 - Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF	CRITICAL	The version of Apache httpd installed on the remote host is equal to or greater than 2.4.7 and prior to 2.4.52. It is, therefore, affected by a flaw related to acting as a forward proxy. A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server Side Request Forgery). Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
11.	153583 - Apache < 2.4.49 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. - A crafted request uri-path can cause mod_proxy to forward the request to an origin server choosen by the remote user. (CVE-2021-40438) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
12.	153584 - Apache < 2.4.49 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.49 changelog. - ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party/external modules may. (CVE-2021-39275) - Malformed requests may cause the server to dereference a NULL pointer. (CVE-2021-34798) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
13.	134162 - PHP 7.2.x < 7.2.28 / PHP 7.3.x < 7.3.15 / 7.4.x < 7.4.3 Multiple Vulnerabilities	CRITICAL	According to its banner, the version of PHP running on the remote web server is either 7.2.x prior to 7.2.28, 7.3.x prior to 7.3.15, or 7.4.x prior to 7.4.3. It is, therefore, affected by multiple vulnerabilities: - A heap-based buffer overflow condition exists in phar_extract_file() function due to incorrect loop termination. An unauthenticated, remote attacker can exploit this to cause a denial of service condition or the execution of arbitrary code. (CVE-2020-7061) - A denial of service (DoS) vulnerability exists in PHP SessionUpload Progress functions due to Null Pointer Dereference. An unauthenticated, remote attacker can exploit this issue to cause the php service to stop responding. (CVE-2020-7062) - An Insecure File Permissions on the buildFromIterator function gives all access permission to Tar files. (CVE-2020-7063)
14.	130276 - PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability	CRITICAL	According to its banner, the version of PHP running on the remote web server is prior to 7.1.33, 7.2.x prior to 7.2.24, or 7.3.x prior to 7.3.11. It is, therefore, affected by a remote code execution vulnerability due to insufficient validation of user input. An unauthenticated, remote attacker can exploit this, by sending a specially crafted request, to cause the execution of arbitrary code by breaking the fastcgi_split_path_info directive.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
15.	58987 - PHP Unsupported Version Detection	CRITICAL	According to its version, the installation of PHP on the remote host is no longer supported. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.
16.	193422 - Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by a http request smuggling vulnerability as referenced in the 2.4.54 advisory. - Possible request smuggling in mod_proxy_ajp: Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.53 and prior versions. Acknowledgements: Ricter Z @ 360 Noah Lab Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
17.	193423 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Denial of Service mod_sed: If Apache HTTP Server 2.4.53 is configured to do transformations with mod_sed in contexts where the input to mod_sed may be very large, mod_sed may make excessively large memory allocations and trigger an abort. Acknowledgements: This issue was found by Brian Moussalli from the JFrog Security Research team Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
18.	193424 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Denial of service in mod_lua r:parsebody: In Apache HTTP Server 2.4.53 and earlier, a malicious request to a lua script that calls r:parsebody(0) may cause a denial of service due to no default limit on possible input size. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-29404) - Information Disclosure in mod_lua with websockets: Apache HTTP Server 2.4.53 and earlier may return lengths to applications calling r:wsread() that point past the end of the storage allocated for the buffer. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-30556) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
19.	183391 - Apache 2.4.x < 2.4.58 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.58. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.58 advisory. - Apache HTTP Server: DoS in HTTP/2 with initial windows size 0: An attacker, opening a HTTP/2 connection with an initial window size of 0, was able to block handling of that connection indefinitely in Apache HTTP Server. This could be used to exhaust worker resources in the server, similar to the well known slow loris attack pattern. This has been fixed in version 2.4.58, so that such connection are terminated properly after the configured connection timeout. This issue affects Apache HTTP Server: from 2.4.55 through 2.4.57. Users are recommended to upgrade to version 2.4.58, which fixes the issue. Acknowledgements: (CVE-2023-43622) - Apache HTTP Server: HTTP/2 stream memory not reclaimed right away on RST: When a HTTP/2 stream was reset (RST frame) by a client, there was a time window were the request's memory resources were not reclaimed immediately. Instead, de-allocation was deferred to connection close. A client could send new requests and resets, keeping the connection busy and open and causing the memory footprint to keep on growing. On connection close, all resources were reclaimed, but the process might run out of memory before that. This was found by the reporter during testing of CVE-2023-44487 (HTTP/2 Rapid Reset Exploit) with their own test client. During normal HTTP/2 use, the probability to hit this bug is very low. The kept memory would not become noticeable before the connection closes or times out. Users are recommended to upgrade to version 2.4.58, which fixes the issue. Acknowledgements: (CVE-2023-45802) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
20.	193419 - Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.58. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.58 advisory. - mod_macro buffer over-read: Out-of-bounds Read vulnerability in mod_macro of Apache HTTP Server. This issue affects Apache HTTP Server: through 2.4.57. Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.
21.	192923 - Apache 2.4.x < 2.4.59 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.59. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.59 advisory. - Apache HTTP Server: HTTP Response Splitting in multiple modules: HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack. Users are recommended to upgrade to version 2.4.59, which fixes this issue. Acknowledgements: (CVE-2024-24795) - Apache HTTP Server: HTTP/2 DoS by memory exhaustion on endless continuation frames: HTTP/2 incoming headers exceeding the limit are temporarily buffered in nghttp2 in order to generate an informative HTTP 413 response. If a client does not stop sending headers, this leads to memory exhaustion. Acknowledgements: finder: Bartek Nowotarski (https://nowotarski.info/) (CVE-2024-27316) Note that Nessus has not tested for these issues but has instead relied only on the application's self reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
22.	153585 - Apache >= 2.4.17 < 2.4.49 mod_http2	HIGH	The version of Apache httpd installed on the remote host is greater than 2.4.17 and prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. A crafted method sent through HTTP/2 will bypass validation and be forwarded by mod_proxy, which can lead to request splitting or cache poisoning. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
23.	153586 - Apache >= 2.4.30 < 2.4.49 mod_proxy_uwsgi	HIGH	The version of Apache httpd installed on the remote host greater than 2.4.30 and is prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. A carefully crafted request uri-path can cause mod_proxy_uwsgi to read above the allocated memory and crash (DoS). Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
24.	140532 - PHP 7.2.x / 7.3.x < 7.3.22 Memory Leak Vulnerability	HIGH	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x or 7.3.x prior to 7.3.21. It is, therefore affected by a memory leak vulnerability in the LDAP component. An unauthenticated, remote attacker could exploit this issue to cause a denial-of-service condition.
25.	129557 - PHP 7.3.x < 7.3.10 Heap-Based Buffer Overflow Vulnerability.	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.10. It is, therefore, affected by a heap-based buffer overflow vulnerability in its mb_eregi component due to insufficient validation of user input. An unauthenticated, remote attacker can exploit this, by sending a specially crafted request, to cause a denial of service condition or the execution of arbitrary code
26.	134944 - PHP 7.3.x < 7.3.16 Multiple Vulnerabilities	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.16. It is, therefore, affected by the following vulnerabilities: - An out of bounds read resulting in the use of an uninitialized value in exif. (CVE-2020-7064)- A stack buffer overflow in mb_strtolower() allows overwriting of a stack-allocated buffer with an overflown array from .rodata. (CVE-2020-7065) - get_headers() silently truncates anything after a null byte in the URL it uses. An unauthenticated, remote attacker can exploit this to leak sensitive information or cause the web server to unexpectedly process attacker-controlled data. (CVE-2020-7066) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
27.	135918 - PHP 7.3.x < 7.3.17 Out of Bounds Read Vulnerability	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.17. It is, therefore, affected by an out-of-bounds read error in its url decoding component due to insufficient validation of user-supplied input. An unauthenticated, remote attacker can exploit this, by sending specially crafted requests, to cause a denial of service (DoS) condition or execution of arbitrary code. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
28.	146311 - PHP 7.3.x < 7.3.27 / 7.4.x < 7.4.15 / 8.x < 8.0.2 DoS	HIGH	The version of PHP installed on the remote host is 7.3.x prior to 7.3.27, 7.4.x prior to 7.4.15, or 8.x prior to 8.0.2. It is, therefore, affected by a denial of service (DoS) vulnerability due to a null dereference in SoapClient. An unauthenticated, remote attacker can exploit this, by providing an XML to the SoapCLient query() function without an existing field, in order to cause PHP to crash. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
29.	154663 - PHP 7.3.x < 7.3.32	HIGH	The version of PHP installed on the remote host is prior to 7.3.32. It is, therefore, affected by a vulnerability as referenced in the Version 7.3.32 advisory. - In PHP versions 7.3.x up to and including 7.3.31, 7.4.x below 7.4.25 and 8.0.x below 8.0.12, when running PHP FPM SAPI with main FPM daemon process running as root and child worker processes running as lower- privileged users, it is possible for the child processes to access memory shared with the main process and write to it, modifying it in a way that would cause the root process to conduct invalid memory reads and writes, which can be used to escalate privileges from local unprivileged user to the root user. (CVE-2021-21703) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
30.	142591 - PHP < 7.3.24 Multiple Vulnerabilities	HIGH	According to its self-reported version number, the version of PHP running on the remote web server is prior to 7.3.24. It is, therefore affected by multiple vulnerabilities
31.	35291 - SSL Certificate Signed Using Weak Hashing Algorithm	HIGH	The remote service uses an SSL certificate chain that has been signed using a cryptographically weak hashing algorithm (e.g. MD2, MD4, MD5, or SHA1). These signature algorithms are known to be vulnerable to collision attacks. An attacker can exploit this to generate another certificate with the same digital signature, allowing an attacker to masquerade as the affected service. Note that this plugin reports all SSL certificate chains signed with SHA-1 that expire after January 1, 2017 as vulnerable. This is in accordance with Google's gradual sunsetting of the SHA-1 cryptographic hash algorithm. Note that certificates in the chain that are contained in the Nessus CA database (known_CA.inc) have been ignored.
32.	135290 - Apache 2.4.x < 2.4.42 Multiple Vulnerabilities	MEDIUM	The version of Apache httpd installed on the remote host is prior to 2.4.42. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.42 advisory. - In Apache HTTP Server 2.4.0 to 2.4.41, mod_proxy_ftp may use uninitialized memory when proxying to a malicious FTP server. (CVE-2020-1934) - In Apache HTTP Server 2.4.0 to 2.4.41, redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL. (CVE-2020-1927) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
33.	193420 - Apache 2.4.x < 2.4.54 Out-Of-Bounds Read (CVE-2022-28330)	MEDIUM	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by an out-of-bounds read vulnerability as referenced in the 2.4.54 advisory. - Read beyond bounds in mod_isapi: Apache HTTP Server 2.4.53 and earlier on Windows may read beyond bounds when configured to process requests with the mod_isapi module. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
34.	128117 - OpenSSL 1.1.0 < 1.1.0l Multiple Vulnerabilities	MEDIUM	The version of OpenSSL installed on the remote host is prior to 1.1.0l. It is, therefore, affected by multiple vulnerabilities as referenced in the 1.1.0l advisory. - In situations where an attacker receives automated notification of the success or failure of a decryption attempt an attacker, after sending a very large number of messages to be decrypted, can recover a CMS/ PKCS7 transported encryption key or decrypt any RSA encrypted message that was encrypted with the public RSA key, using a Bleichenbacher padding oracle attack. Applications are not affected if they use a certificate together with the private RSA key to the CMS_decrypt or PKCS7_decrypt functions to select the correct recipient info to decrypt. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1563) - Normally in OpenSSL EC groups always have a co-factor present and this is used in side channel resistant code paths. However, in some cases, it is possible to construct a group using explicit parameters (instead of using a named curve). In those cases it is possible that such a group does not have the cofactor present. This can occur even where all the parameters match a known named curve. If such a curve is used then OpenSSL falls back to non-side channel resistant code paths which may result in full key recovery during an ECDSA signature operation. In order to be vulnerable an attacker would have to have the ability to time the creation of a large number of signatures where explicit parameters with no co-factor present are in use by an application using libcrypto. For the avoidance of doubt libssl is not vulnerable because explicit parameters are never used. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1547) - OpenSSL has internal defaults for a directory tree where it can find a configuration file as well as certificates used for verification in TLS. This directory is most commonly referred to as OPENSSLDIR, and is configurable with the --prefix / --openssldir configuration options. For OpenSSL versions 1.1.0 and 1.1.1, the mingw configuration targets assume that resulting programs and libraries are installed in a Unix-like environment and the default prefix for program installation as well as for OPENSSLDIR should be '/usr/local'. However, mingw programs are Windows programs, and as such, find themselves looking at sub- directories of 'C:/usr/local', which may be world writable, which enables untrusted users to modify OpenSSL's default configuration, insert CA certificates, modify (or even replace) existing engine modules, etc. For OpenSSL 1.0.2, '/usr/local/ssl' is used as default for OPENSSLDIR on all Unix and Windows targets, including Visual C builds. However, some build instructions for the diverse Windows targets on 1.0.2 encourage you to specify your own --prefix. OpenSSL versions 1.1.1, 1.1.0 and 1.0.2 are affected by this issue. Due to the limited scope of affected deployments this has been assessed as low severity and therefore we are not creating new releases at this time. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1552) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
35.	141355 - PHP 7.2 < 7.2.34 / 7.3.x < 7.3.23 / 7.4.x < 7.4.11 Mulitiple Vulnerabilities	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x prior to 7.2.34, 7.3.x prior to 7.3.23 or 7.4.x prior to 7.4.11. It is, therefore, affected by multiple vulnerabilties: - A weak cryptography vulnerability exists in PHP's openssl_encrypt function due to a failure to utilize all provided IV bytes. An unauthenticated, remote attacker could exploit this to reduce the level of security provided by the encryption scheme or affect the integrity of the encrypted data (CVE-2020-7069). - A cookie forgery vulnerability exists in PHP's HTTP processing functionality. An unauthenticated, remote could expoit this to forge HTTP cookies which were supposed to be secure. (CVE-2020-7070) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
36.	136741 - PHP 7.2.x < 7.2.31 / 7.3.x < 7.3.18, 7.4.x < 7.4.6 Denial of Service (DoS)	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x prior to 7.2.31, 7.3.x prior to 7.3.18 or 7.4.x prior to 7.4.6. It is, therefore, affected by a denial of service (DoS) vulnerability in its HTTP file upload component due to a failure to clean up temporary files created during the file upload process. An unauthenticated, remote attacker can exploit this issue, by repeatedly submitting uploads with long file or field names, to exhaust disk space and cause a DoS condition. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
37.	143449 - PHP 7.3.x < 7.3.25 / 7.4.x < 7.4.13 Multiple Vulnerabilities	MEDIUM	The version of PHP installed on the remote host is 7.3.x prior to 7.3.25 or 7.4.x prior to 7.4.13. It is, therefore, affected by multiple vulnerabilities as specified by the changelogs of the respective fixed releases. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
38.	144947 - PHP 7.3.x < 7.3.26 / 7.4.x < 7.4.14 / 8.x < 8.0.1 Input Validation Error	MEDIUM	The version of PHP installed on the remote host is 7.3.x prior to 7.3.26, 7.4.x prior to 7.4.14, or 8.x prior to 8.0.1. It is, therefore, affected by an input validation error due to insufficient validation of a URL, as specified by the changelogs of the respective fixed releases. An unauthenticated, remote attacker can exploit this, by including an '@' character, in order to bypass the URL filter. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
39.	155590 - PHP 7.3.x < 7.3.33	MEDIUM	The version of PHP installed on the remote host is prior to 7.3.33. It is, therefore, affected by a vulnerability as referenced in the Version 7.3.33 advisory. - In PHP versions 7.3.x below 7.3.33, 7.4.x below 7.4.26 and 8.0.x below 8.0.13, certain XML parsing functions, like simplexml_load_file(), URL-decode the filename passed to them. If that filename contains URL-encoded NUL character, this may cause the function to interpret this as the end of the filename, thus interpreting the filename differently from what the user intended, which may lead it to reading a different file than intended. (CVE-2021-21707) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-1 (ต่อ)

No.	Name	Level	Descriptiom
40.	152853 - PHP < 7.3.28 Email Header Injection	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is prior to 7.3.28. It is, therefore affected by an email header injection vulnerability, due to a failure to properly handle CR LF sequences in header fields. An unauthenticated, remote attacker can exploit this, by inserting line feed characters into email headers, to gain full control of email header content.
41.	51192 - SSL Certificate Cannot Be Trusted	MEDIUM	The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below : - First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority. - Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates. - Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize. If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the middle attacks against the remote host.
42.	15901 - SSL Certificate Expiry	MEDIUM	This plugin checks expiry dates of certificates associated with SSL- enabled services on the target and reports whether any have already expired.
43.	57582 - SSL Self-Signed Certificate	MEDIUM	The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the middle attack against the remote host. Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.
44.	104743 - TLS Version 1.0 Protocol Detection	MEDIUM	The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible. As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors. PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.
45.	157288 - TLS Version 1.1 Deprecated Protocol	MEDIUM	The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1 As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

จากการสแกน Moodle 3.7.1 ก่อนดำเนินการแก้ไข พบช่องโหว่ด้านความมั่นคงปลอดภัยรวม 45 รายการ โดยแบ่งเป็น CRITICAL 15 รายการ, HIGH 16 รายการ และ MEDIUM 14 รายการ ตามที่แสดงใน ตารางที่ ค-1 ช่องโหว่ที่พบส่วนใหญ่เกี่ยวข้องกับ Apache HTTP Server, PHP, OpenSSL และโปรโตคอล TLS ที่ล้าสมัย ซึ่งอาจเปิดโอกาสให้เกิดการโจมตี เช่น Remote Code Execution (RCE), Denial of Service (DoS), HTTP Request Smuggling และ Authentication Bypass นอกจากนี้ยังพบปัญหาด้าน SSL Certificate และการใช้ TLS 1.0/1.1 ที่ไม่ปลอดภัย

เพื่อป้องกันความเสี่ยง ควร อัปเดต Apache เป็นเวอร์ชันล่าสุด, อัปเดต PHP เป็นเวอร์ชัน 8.x, ปิดใช้งาน TLS 1.0 และ 1.1, ติดตั้งใบรับรอง SSL ที่ได้รับการรับรอง และกำหนดค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ให้เหมาะสม รวมถึงดำเนินการ re-scan หลังจากแก้ไขเพื่อยืนยันว่าช่องโหว่ได้รับการแก้ไขเรียบร้อยแล้ว ระบบยังคงมีความเสี่ยงสูงและจำเป็นต้องดำเนินการปรับปรุงเพิ่มเติมเพื่อลดช่องโหว่ที่เหลืออยู่

ตารางที่ ค-2 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก Nessus ก่อนการแก้ไขช่องโหว่

No.	Name	Level	Descrptiom
1.	51192 - SSL Certificate Cannot Be Trusted	MEDIUM	The server's X.509 certificate cannot be trusted. This situation can occur in three different ways, in which the chain of trust can be broken, as stated below : - First, the top of the certificate chain sent by the server might not be descended from a known public certificate authority. This can occur either when the top of the chain is an unrecognized, self-signed certificate, or when intermediate certificates are missing that would connect the top of the certificate chain to a known public certificate authority.- Second, the certificate chain may contain a certificate that is not valid at the time of the scan. This can occur either when the scan occurs before one of the certificate's 'notBefore' dates, or after one of the certificate's 'notAfter' dates. - Third, the certificate chain may contain a signature that either didn't match the certificate's information or could not be verified. Bad signatures can be fixed by getting the certificate with the bad signature to be re-signed by its issuer. Signatures that could not be verified are the result of the certificate's issuer using a signing algorithm that Nessus either does not support or does not recognize. If the remote host is a public host in production, any break in the chain makes it more difficult for users to verify the authenticity and identity of the web server. This could make it easier to carry out man-in-the middle attacks against the remote host.
2.	57582 - SSL Self-Signed Certificate	MEDIUM	The X.509 certificate chain for this service is not signed by a recognized certificate authority. If the remote host is a public host in production, this nullifies the use of SSL as anyone could establish a man-in-the middle attack against the remote host. Note that this plugin does not check for certificate chains that end in a certificate that is not self-signed, but is signed by an unrecognized certificate authority.

จากการสแกน Moodle 4.5.1 ก่อนการแก้ไข พบว่าระบบยังคงมีช่องโหว่ด้านความมั่นคงปลอดภัยที่ต้องดำเนินการปรับปรุง โดยพบปัญหาหลักเกี่ยวกับ SSL Certificate ซึ่งรวมถึง ใบรับรองที่ไม่น่าเชื่อถือ (Untrusted SSL Certificate) และการใช้ใบรับรองที่ลงนามเอง (Self-Signed Certificate) ตามที่แสดงใน ตารางที่ ค-2 ช่องโหว่เหล่านี้อาจทำให้เกิดความเสี่ยงต่อ Man-in-the-Middle (MitM) Attack และส่งผลให้ผู้ใช้งานไม่สามารถตรวจสอบความน่าเชื่อถือของเว็บไซต์ได้ จึงควรได้รับการแก้ไขโดย ติดตั้งใบรับรอง SSL จาก CA ที่เชื่อถือได้ และตรวจสอบโซ่ใบรับรอง (Certificate Chain) ให้ถูกต้อง เพื่อลดความเสี่ยงด้านความมั่นคงปลอดภัย และเพิ่มความน่าเชื่อถือของระบบ

ตารางที่ ค-3 รายการช่องโหว่ของ Moodle 3.7.1 ที่ตรวจพบจาก Nessus หลังการแก้ไขช่องโหว่

No.	Name	Level	Descrptiom
1.	139574 - Apache 2.4.x < 2.4.46 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.46. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.46 advisory.- Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE (CVE-2020-11984) - Apache HTTP Server versions 2.4.20 to 2.4.43 When trace/debug was enabled for the HTTP/2 module and on certain traffic edge patterns, logging statements were made on the wrong connection, causing concurrent use of memory pools. Configuring the LogLevel of mod_http2 above info will mitigate this vulnerability for unpatched servers. (CVE-2020-11993) - Apache HTTP Server versions 2.4.20 to 2.4.43. A specially crafted value for the 'Cache-Digest' header in a HTTP/2 request would result in a crash when the server actually tries to HTTP/2 PUSH a resource afterwards. Configuring the HTTP/2 feature via H2Push off will mitigate this vulnerability for unpatched servers. (CVE-2020-9490) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
2.	150280 - Apache 2.4.x < 2.4.47 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.47. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.47 changelog: - Unexpected <Location> section matching with 'MergeSlashes OFF' (CVE-2021-30641) - mod_auth_digest: possible stack overflow by one nul byte while validating the Digest nonce. (CVE-2020-35452) - mod_session: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service with a malicious backend server and SessionHeader. (CVE-2021-26691) - mod_session: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service. (CVE-2021-26690) - mod_proxy_http: Fix possible crash due to NULL pointer dereference, which could be used to cause a Denial of Service. (CVE-2020-13950) - Windows: Prevent local users from stopping the httpd process (CVE-2020-13938) - mod_proxy_wstunnel, mod_proxy_http: Handle Upgradable protocols end-to-end negotiation. (CVE-2019-17567) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
3.	161454 - Apache 2.4.x < 2.4.52 mod_lua Buffer Overflow	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.52. It is, therefore, affected by a flaw related to mod_lua when handling multipart content. A carefully crafted request body can cause a buffer overflow in the mod_lua multipart parser (r:parsebody() called from Lua scripts). The Apache httpd team is not aware of an exploit for the vulnerability though it might be possible to craft one. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
4.	158900 - Apache 2.4.x < 2.4.53 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.53. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.53 advisory. - mod_lua Use of uninitialized value of in r:parsebody: A carefully crafted request body can cause a read to a random memory area which could cause the process to crash. This issue affects Apache HTTP Server 2.4.52 and earlier. Acknowledgements: Chamal De Silva (CVE-2022-22719) - HTTP request smuggling: Apache HTTP Server 2.4.52 and earlier fails to close inbound connection when errors are encountered discarding the request body, exposing the server to HTTP Request Smuggling Acknowledgements: James Kettle <james.kettle portswigger.net> (CVE-2022-22720) - Possible buffer overflow with very large or unlimited LimitXMLRequestBody in core: If LimitXMLRequestBody is set to allow request bodies larger than 350MB (defaults to 1M) on 32 bit systems an integer overflow happens which later causes out of bounds writes. This issue affects Apache HTTP Server 2.4.52 and earlier. Acknowledgements: Anonymous working with Trend Micro Zero Day Initiative (CVE-2022-22721) - Read/write beyond bounds in mod_sed: Out-of-bounds Write vulnerability in mod_sed of Apache HTTP Server allows an attacker to overwrite heap memory with possibly attacker provided data. This issue affects Apache HTTP Server 2.4 version 2.4.52 and prior versions. Acknowledgements: Ronald Crane (Zippenhop LLC) (CVE-2022-23943) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
5.	193421 - Apache 2.4.x < 2.4.54 Authentication Bypass	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by an authentication bypass vulnerability as referenced in the 2.4.54 advisory. - X-Forwarded-For dropped by hop-by-hop mechanism in mod_proxy: Apache HTTP Server 2.4.53 and earlier may not send the X-Forwarded-* headers to the origin server based on client side Connection header hop-by-hop mechanism. This may be used to bypass IP based authentication on the origin server/application. Acknowledgements: The Apache HTTP Server project would like to thank Gaetan Ferry (Synacktiv) for reporting this issue Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
6.	161948 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Read beyond bounds via ap_rwrite(): The ap_rwrite() function in Apache HTTP Server 2.4.53 and earlier may read unintended memory if an attacker can cause the server to reflect very large input using ap_rwrite() or ap_rputs(), such as with mod_luas r:puts() function. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28614) - Read beyond bounds in ap_strcmp_match(): Apache HTTP Server 2.4.53 and earlier may crash or disclose information due to a read beyond bounds in ap_strcmp_match() when provided with an extremely large input buffer. While no code distributed with the server can be coerced into such a call, third-party modules or lua scripts that use ap_strcmp_match() may hypothetically be affected. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-28615) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
7.	170113 - Apache 2.4.x < 2.4.55 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.55. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.55 advisory. - A carefully crafted If: request header can cause a memory read, or write of a single zero byte, in a pool (heap) memory location beyond the header value sent. This could cause the process to crash. This issue affects Apache HTTP Server 2.4.54 and earlier. (CVE-2006-20001) - Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.54 and prior versions. (CVE-2022-36760) - Prior to Apache HTTP Server 2.4.55, a malicious backend can cause the response headers to be truncated early, resulting in some headers being incorporated into the response body. If the later headers have any security purpose, they will not be interpreted by the client. (CVE-2022-37436) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
8.	172186 - Apache 2.4.x < 2.4.56 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.56. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.56 advisory. - HTTP request splitting with mod_rewrite and mod_proxy: Some mod_proxy configurations on Apache HTTP Server versions 2.4.0 through 2.4.55 allow a HTTP Request Smuggling attack. Configurations are affected when mod_proxy is enabled along with some form of RewriteRule or ProxyPassMatch in which a non-specific pattern matches some portion of the user-supplied request-target (URL) data and is then re-inserted into the proxied request-target using variable substitution. For example, something like: RewriteEngine on RewriteRule ^/here/(.*) http://example.com:8080/elsewhere?$1 http://example.com:8080/elsewhere ; [P] ProxyPassReverse /here/ http://example.com:8080/ http:// example.com:8080/ Request splitting/smuggling could result in bypass of access controls in the proxy server, proxying unintended URLs to existing origin servers, and cache poisoning. Acknowledgements: finder: Lars Krapf of Adobe (CVE-2023-25690) - Apache HTTP Server: mod_proxy_uwsgi HTTP response splitting: HTTP Response Smuggling vulnerability in Apache HTTP Server via mod_proxy_uwsgi. This issue affects Apache HTTP Server: from 2.4.30 through 2.4.55. Special characters in the origin response header can truncate/split the response forwarded to the client. Acknowledgements: finder: Dimas Fariski Setyawan Putra (nyxsorcerer) (CVE-2023-27522) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
9.	201198 - Apache 2.4.x < 2.4.60 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.60. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.60 advisory. - Serving WebSocket protocol upgrades over a HTTP/2 connection could result in a Null Pointer dereference, leading to a crash of the server process, degrading performance. (CVE-2024-36387) - SSRF in Apache HTTP Server on Windows allows to potentially leak NTML hashes to a malicious server via SSRF and malicious requests or content Users are recommended to upgrade to version 2.4.60 which fixes this issue. Note: Existing configurations that access UNC paths will have to configure new directive UNCList to allow access during request processing. (CVE-2024-38472) - Encoding problem in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows request URLs with incorrect encoding to be sent to backend services, potentially bypassing authentication via crafted requests. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38473) - Substitution encoding issue in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows attacker to execute scripts in directories permitted by the configuration but not directly reachable by any URL or source disclosure of scripts meant to only to be executed as CGI. Users are recommended to upgrade to version 2.4.60, which fixes this issue. Some RewriteRules that capture and substitute unsafely will now fail unless rewrite flag UnsafeAllow3F is specified. (CVE-2024-38474) - Improper escaping of output in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to map URLs to filesystem locations that are permitted to be served by the server but are not intentionally/directly reachable by any URL, resulting in code execution or source code disclosure. Substitutions in server context that use a backreferences or variables as the first segment of the substitution are affected. Some unsafe RewiteRules will be broken by this change and the rewrite flag UnsafePrefixStat can be used to opt back in once ensuring the substitution is appropriately constrained. (CVE-2024-38475) - Vulnerability in core of Apache HTTP Server 2.4.59 and earlier are vulnerably to information disclosure, SSRF or local script execution via backend applications whose response headers are malicious or exploitable. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38476)- null pointer dereference in mod_proxy in Apache HTTP Server 2.4.59 and earlier allows an attacker to crash the server via a malicious request. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-38477) - Potential SSRF in mod_rewrite in Apache HTTP Server 2.4.59 and earlier allows an attacker to cause unsafe RewriteRules to unexpectedly setup URL's to be handled by mod_proxy. Users are recommended to upgrade to version 2.4.60, which fixes this issue. (CVE-2024-39573) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
10.	156255 - Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF	CRITICAL	The version of Apache httpd installed on the remote host is equal to or greater than 2.4.7 and prior to 2.4.52. It is, therefore, affected by a flaw related to acting as a forward proxy. A crafted URI sent to httpd configured as a forward proxy (ProxyRequests on) can cause a crash (NULL pointer dereference) or, for configurations mixing forward and reverse proxy declarations, can allow for requests to be directed to a declared Unix Domain Socket endpoint (Server Side Request Forgery). Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
11.	153583 - Apache < 2.4.49 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. - A crafted request uri-path can cause mod_proxy to forward the request to an origin server choosen by the remote user. (CVE-2021-40438) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
12.	53584 - Apache < 2.4.49 Multiple Vulnerabilities	CRITICAL	The version of Apache httpd installed on the remote host is prior to 2.4.49. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.49 changelog. - ap_escape_quotes() may write beyond the end of a buffer when given malicious input. No included modules pass untrusted data to these functions, but third-party / external modules may. (CVE-2021-39275) - Malformed requests may cause the server to dereference a NULL pointer. (CVE-2021-34798) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
13.	134162 - PHP 7.2.x < 7.2.28 / PHP 7.3.x < 7.3.15 / 7.4.x < 7.4.3 Multiple Vulnerabilities	CRITICAL	According to its banner, the version of PHP running on the remote web server is either 7.2.x prior to 7.2.28, 7.3.x prior to 7.3.15, or 7.4.x prior to 7.4.3. It is, therefore, affected by multiple vulnerabilities: - A heap-based buffer overflow condition exists in phar_extract_file() function due to incorrect loop termination. An unauthenticated, remote attacker can exploit this to cause a denial of service condition or the execution of arbitrary code. (CVE-2020-7061) - A denial of service (DoS) vulnerability exists in PHP SessionUpload Progress functions due to Null Pointer Dereference. An unauthenticated, remote attacker can exploit this issue to cause the php service to stop responding. (CVE-2020-7062) - An Insecure File Permissions on the buildFromIterator function gives all access permission to Tar files. (CVE-2020-7063)

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
14.	130276 - PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.	CRITICAL	According to its banner, the version of PHP running on the remote web server is prior to 7.1.33, 7.2.x prior to 7.2.24, or 7.3.x prior to 7.3.11. It is, therefore, affected by a remote code execution vulnerability due to insufficient validation of user input. An unauthenticated, remote attacker can exploit this, by sending a specially crafted request, to cause the execution of arbitrary code by breaking the fastcgi_split_path_info directive.
15.	58987 - PHP Unsupported Version Detection	CRITICAL	According to its version, the installation of PHP on the remote host is no longer supported. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.
16.	193422 - Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by a http request smuggling vulnerability as referenced in the 2.4.54 advisory. - Possible request smuggling in mod_proxy_ajp: Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling') vulnerability in mod_proxy_ajp of Apache HTTP Server allows an attacker to smuggle requests to the AJP server it forwards requests to. This issue affects Apache HTTP Server Apache HTTP Server 2.4 version 2.4.53 and prior versions. Acknowledgements: Ricter Z @ 360 Noah Lab Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
17.	193423 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Denial of Service mod_sed: If Apache HTTP Server 2.4.53 is configured to do transformations with mod_sed in contexts where the input to mod_sed may be very large, mod_sed may make excessively large memory allocations and trigger an abort. Acknowledgements: This issue was found by Brian Moussalli from the JFrog Security Research team Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
18.	193424 - Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.54 advisory. - Denial of service in mod_lua r:parsebody: In Apache HTTP Server 2.4.53 and earlier, a malicious request to a lua script that calls r:parsebody(0) may cause a denial of service due to no default limit on possible input size. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-29404) - Information Disclosure in mod_lua with websockets: Apache HTTP Server 2.4.53 and earlier may return lengths to applications calling r:wsread() that point past the end of the storage allocated for the buffer. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue (CVE-2022-30556) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
19.	183391 - Apache 2.4.x < 2.4.58 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.58. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.58 advisory. - Apache HTTP Server: DoS in HTTP/2 with initial windows size 0: An attacker, opening a HTTP/2 connection with an initial window size of 0, was able to block handling of that connection indefinitely in Apache HTTP Server. This could be used to exhaust worker resources in the server, similar to the well known slow loris attack pattern. This has been fixed in version 2.4.58, so that such connection are terminated properly after the configured connection timeout. This issue affects Apache HTTP Server: from 2.4.55 through 2.4.57. Users are recommended to upgrade to version 2.4.58, which fixes the issue. Acknowledgements: (CVE-2023-43622) - Apache HTTP Server: HTTP/2 stream memory not reclaimed right away on RST: When a HTTP/2 stream was reset (RST frame) by a client, there was a time window were the request's memory resources were not reclaimed immediately. Instead, de-allocation was deferred to connection close. A client could send new requests and resets, keeping the connection busy and open and causing the memory footprint to keep on growing. On connection close, all resources were reclaimed, but the process might run out of memory before that. This was found by the reporter during testing of CVE-2023-44487 (HTTP/2 Rapid Reset Exploit) with their own test client. During normal HTTP/2 use, the probability to hit this bug is very low. The kept memory would not become noticeable before the connection closes or times out. Users are recommended to upgrade to version 2.4.58, which fixes the issue. Acknowledgements: (CVE-2023-45802) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
20.	193419 - Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.58. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.58 advisory. - mod_macro buffer over-read: Out-of-bounds Read vulnerability in mod_macro of Apache HTTP Server. This issue affects Apache HTTP Server: through 2.4.57. Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
21.	192923 - Apache 2.4.x < 2.4.59 Multiple Vulnerabilities	HIGH	The version of Apache httpd installed on the remote host is prior to 2.4.59. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.59 advisory. - Apache HTTP Server: HTTP Response Splitting in multiple modules: HTTP Response splitting in multiple modules in Apache HTTP Server allows an attacker that can inject malicious response headers into backend applications to cause an HTTP desynchronization attack. Users are recommended to upgrade to version 2.4.59, which fixes this issue. Acknowledgements: (CVE-2024-24795) - Apache HTTP Server: HTTP/2 DoS by memory exhaustion on endless continuation frames: HTTP/2 incoming headers exceeding the limit are temporarily buffered in nghttp2 in order to generate an informative HTTP 413 response. If a client does not stop sending headers, this leads to memory exhaustion. Acknowledgements: finder: Bartek Nowotarski (https://nowotarski.info/) (CVE-2024-27316) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
22.	153585 - Apache >= 2.4.17 < 2.4.49 mod_http2	HIGH	The version of Apache httpd installed on the remote host is greater than 2.4.17 and prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. A crafted method sent through HTTP/2 will bypass validation and be forwarded by mod_proxy, which can lead to request splitting or cache poisoning. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
23.	153586 - Apache >= 2.4.30 < 2.4.49 mod_proxy_uwsgi	HIGH	The version of Apache httpd installed on the remote host greater than 2.4.30 and is prior to 2.4.49. It is, therefore, affected by a vulnerability as referenced in the 2.4.49 changelog. A carefully crafted request uri-path can cause mod_proxy_uwsgi to read above the allocated memory and crash (DoS). Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
24.	140532 - PHP 7.2.x / 7.3.x < 7.3.22 Memory Leak Vulnerability	HIGH	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x or 7.3.x prior to 7.3.21. It is, therefore affected by a memory leak vulnerability in the LDAP component. An unauthenticated, remote attacker could exploit this issue to cause a denial-of-service condition.
25.	129557 - PHP 7.3.x < 7.3.10 Heap-Based Buffer Overflow Vulnerability.	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.10. It is, therefore, affected by a heap-based buffer overflow vulnerability in its mb_eregi component due to insufficient validation of user input. An unauthenticated, remote attacker can exploit this, by sending a specially crafted request, to cause a denial of service condition or the execution of arbitrary code

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
26.	134944 - PHP 7.3.x < 7.3.16 Multiple Vulnerabilities	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.16. It is, therefore, affected by the following vulnerabilities: - An out of bounds read resulting in the use of an uninitialized value in exif. (CVE-2020-7064) - A stack buffer overflow in mb_strtolower() allows overwriting of a stack-allocated buffer with an overflown array from .rodata. (CVE-2020-7065) - get_headers() silently truncates anything after a null byte in the URL it uses. An unauthenticated, remote attacker can exploit this to leak sensitive information or cause the web server to unexpectedly process attacker-controlled data. (CVE-2020-7066) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
27.	135918 - PHP 7.3.x < 7.3.17 Out of Bounds Read Vulnerability	HIGH	According to its banner, the version of PHP running on the remote web server is 7.3.x prior to 7.3.17. It is, therefore, affected by an out-of-bounds read error in its url decoding component due to insufficient validation of user-supplied input. An unauthenticated, remote attacker can exploit this, by sending specially crafted requests, to cause a denial of service (DoS) condition or execution of arbitrary code. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
28.	146311 - PHP 7.3.x < 7.3.27 / 7.4.x < 7.4.15 / 8.x < 8.0.2 DoS	HIGH	The version of PHP installed on the remote host is 7.3.x prior to 7.3.27, 7.4.x prior to 7.4.15, or 8.x prior to 8.0.2. It is, therefore, affected by a denial of service (DoS) vulnerability due to a null dereference in SoapClient. An unauthenticated, remote attacker can exploit this, by providing an XML to the SoapCLient query() function without an existing field, in order to cause PHP to crash. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
29.	154663 - PHP 7.3.x < 7.3.32	HIGH	The version of PHP installed on the remote host is prior to 7.3.32. It is, therefore, affected by a vulnerability as referenced in the Version 7.3.32 advisory. - In PHP versions 7.3.x up to and including 7.3.31, 7.4.x below 7.4.25 and 8.0.x below 8.0.12, when running PHP FPM SAPI with main FPM daemon process running as root and child worker processes running as lower- privileged users, it is possible for the child processes to access memory shared with the main process and write to it, modifying it in a way that would cause the root process to conduct invalid memory reads and writes, which can be used to escalate privileges from local unprivileged user to the root user. (CVE-2021-21703) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
30.	142591 - PHP < 7.3.24 Multiple Vulnerabilities	HIGH	According to its self-reported version number, the version of PHP running on the remote web server is prior to 7.3.24. It is, therefore affected by multiple vulnerabilities
31.	135290 - Apache 2.4.x < 2.4.42 Multiple Vulnerabilities	MEDIUM	The version of Apache httpd installed on the remote host is prior to 2.4.42. It is, therefore, affected by multiple vulnerabilities as referenced in the 2.4.42 advisory. - In Apache HTTP Server 2.4.0 to 2.4.41, mod_proxy_ftp may use uninitialized memory when proxying to a malicious FTP server. (CVE-2020-1934) - In Apache HTTP Server 2.4.0 to 2.4.41, redirects configured with mod_rewrite that were intended to be self-referential might be fooled by encoded newlines and redirect instead to an unexpected URL within the request URL. (CVE-2020-1927) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
32.	193420 - Apache 2.4.x < 2.4.54 Out-Of-Bounds Read (CVE-2022-28330)	MEDIUM	The version of Apache httpd installed on the remote host is prior to 2.4.54. It is, therefore, affected by an out-of-bounds read vulnerability as referenced in the 2.4.54 advisory. - Read beyond bounds in mod_isapi: Apache HTTP Server 2.4.53 and earlier on Windows may read beyond bounds when configured to process requests with the mod_isapi module. Acknowledgements: The Apache HTTP Server project would like to thank Ronald Crane (Zippenhop LLC) for reporting this issue Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
33.	128117 - OpenSSL 1.1.0 < 1.1.0l Multiple Vulnerabilities	MEDIUM	The version of OpenSSL installed on the remote host is prior to 1.1.0l. It is, therefore, affected by multiple vulnerabilities as referenced in the 1.1.0l advisory. - In situations where an attacker receives automated notification of the success or failure of a decryption attempt an attacker, after sending a very large number of messages to be decrypted, can recover a CMS/ PKCS7 transported encryption key or decrypt any RSA encrypted message that was encrypted with the public RSA key, using a Bleichenbacher padding oracle attack. Applications are not affected if they use a certificate together with the private RSA key to the CMS_decrypt or PKCS7_decrypt functions to select the correct recipient info to decrypt. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1563) - Normally in OpenSSL EC groups always have a co-factor present and this is used in side channel resistant code paths. However, in some cases, it is possible to construct a group using explicit parameters (instead of using a named curve). In those cases it is possible that such a group does not have the cofactor present. This can occur even where all the parameters match a known named curve. If such a curve is used then OpenSSL falls back to non-side channel resistant code paths which may result in full key recovery during an ECDSA signature operation. In order to be vulnerable an attacker would have to have the ability to time the creation of a large number of signatures where explicit parameters with no co-factor present are in use by an

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
			application using libcrypto. For the avoidance of doubt libssl is not vulnerable because explicit parameters are never used. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1547) - OpenSSL has internal defaults for a directory tree where it can find a configuration file as well as certificates used for verification in TLS. This directory is most commonly referred to as OPENSSLDIR, and is configurable with the --prefix / --openssldir configuration options. For OpenSSL versions 1.1.0 and 1.1.1, the mingw configuration targets assume that resulting programs and libraries are installed in a Unix-like environment and the default prefix for program installation as well as for OPENSSLDIR should be '/usr/local'. However, mingw programs are Windows programs, and as such, find themselves looking at sub- directories of 'C:/usr/local', which may be world writable, which enables untrusted users to modify OpenSSL's default configuration, insert CA certificates, modify (or even replace) existing engine modules, etc. For OpenSSL 1.0.2, '/usr/local/ssl' is used as default for OPENSSLDIR on all Unix and Windows targets, including Visual C builds. However, some build instructions for the diverse Windows targets on 1.0.2 encourage you to specify your own --prefix. OpenSSL versions 1.1.1, 1.1.0 and 1.0.2 are affected by this issue. Due to the limited scope of affected deployments this has been assessed as low severity and therefore we are not creating new releases at this time. Fixed in OpenSSL 1.1.1d (Affected 1.1.1-1.1.1c). Fixed in OpenSSL 1.1.0l (Affected 1.1.0-1.1.0k). Fixed in OpenSSL 1.0.2t (Affected 1.0.2-1.0.2s). (CVE-2019-1552) Note that Nessus has not tested for these issues but has instead relied only on the application's self-reported version number.
34.	141355 - PHP 7.2 < 7.2.34 / 7.3.x < 7.3.23 / 7.4.x < 7.4.11 Mulitiple Vulnerabilities	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x prior to 7.2.34, 7.3.x prior to 7.3.23 or 7.4.x prior to 7.4.11. It is, therefore, affected by multiple vulnerabilties: - A weak cryptography vulnerability exists in PHP's openssl_encrypt function due to a failure to utilize all provided IV bytes. An unauthenticated, remote attacker could exploit this to reduce the level of security provided by the encryption scheme or affect the integrity of the encrypted data (CVE-2020-7069). - A cookie forgery vulnerability exists in PHP's HTTP processing functionality. An unauthenticated, remote could expoit this to forge HTTP cookies which were supposed to be secure. (CVE-2020-7070) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
35.	136741 - PHP 7.2.x < 7.2.31 / 7.3.x < 7.3.18, 7.4.x < 7.4.6 Denial of Service (DoS)	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is 7.2.x prior to 7.2.31, 7.3.x prior to 7.3.18 or 7.4.x prior to 7.4.6. It is, therefore, affected by a denial of service (DoS) vulnerability in its HTTP file upload component due to a failure to clean up temporary files created during the file upload process. An unauthenticated, remote attacker can exploit this issue, by repeatedly submitting uploads with long file or field names, to exhaust disk space and cause a DoS condition.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
			Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
36.	143449 - PHP 7.3.x < 7.3.25 / 7.4.x < 7.4.13 Multiple Vulnerabilities	MEDIUM	The version of PHP installed on the remote host is 7.3.x prior to 7.3.25 or 7.4.x prior to 7.4.13. It is, therefore, affected by multiple vulnerabilities as specified by the changelogs of the respective fixed releases. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
37.	144947 - PHP 7.3.x < 7.3.26 / 7.4.x < 7.4.14 / 8.x < 8.0.1 Input Validation Error	MEDIUM	The version of PHP installed on the remote host is 7.3.x prior to 7.3.26, 7.4.x prior to 7.4.14, or 8.x prior to 8.0.1. It is, therefore, affected by an input validation error due to insufficient validation of a URL, as specified by the changelogs of the respective fixed releases. An unauthenticated, remote attacker can exploit this, by including an '@' character, in order to bypass the URL filter. Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version
38.	155590 - PHP 7.3.x < 7.3.33	MEDIUM	The version of PHP installed on the remote host is prior to 7.3.33. It is, therefore, affected by a vulnerability as referenced in the Version 7.3.33 advisory. - In PHP versions 7.3.x below 7.3.33, 7.4.x below 7.4.26 and 8.0.x below 8.0.13, certain XML parsing functions, like simplexml_load_file(), URL-decode the filename passed to them. If that filename contains URL-encoded NUL character, this may cause the function to interpret this as the end of the filename, thus interpreting the filename differently from what the user intended, which may lead it to reading a different file than intended. (CVE-2021-21707) Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.
39.	152853 - PHP < 7.3.28 Email Header Injection	MEDIUM	According to its self-reported version number, the version of PHP running on the remote web server is prior to 7.3.28. It is, therefore affected by an email header injection vulnerability, due to a failure to properly handle CR-LF sequences in header fields. An unauthenticated, remote attacker can exploit this, by inserting line feed characters into email headers, to gain full control of email header content.
40.	104743 - TLS Version 1.0 Protocol Detection	MEDIUM	The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible. As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors. PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

ตารางที่ ค-3 (ต่อ)

No.	Name	Level	Descriptiom
41.	157288 - TLS Version 1.1 Deprecated Protocol	MEDIUM	The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1 As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

จากการสแกน Moodle 3.7.1 หลังการแก้ไข พบช่องโหว่รวม 41 รายการ แบ่งเป็น CRITICAL 15 รายการ HIGH 15 รายการ และ MEDIUM 11 รายการ ตามที่แสดงใน ตารางที่ ค-3 โดยส่วนใหญ่เกี่ยวข้องกับ Apache HTTP Server, PHP, OpenSSL และการใช้โปรโตคอลเข้ารหัสที่ล้าสมัย (TLS 1.0 และ 1.1) ซึ่งอาจนำไปสู่การโจมตีแบบ Remote Code Execution (RCE), Authentication Bypass และ Denial of Service (DoS) เพื่อเพิ่มความมั่นคงปลอดภัยควรอัปเดต Apache เป็นเวอร์ชัน 2.4.60 ขึ้นไป อัปเดต PHP เป็นเวอร์ชัน 8.x ปิดใช้งาน TLS 1.0 และ 1.1 และปรับแต่งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์ พร้อมทั้งดำเนินการ re-scan เพื่อยืนยันการแก้ไขช่องโหว่ ระบบยังมีความเสี่ยงสูงและต้องดำเนินการปรับปรุงเพิ่มเติมเพื่อลดความเสี่ยงด้านความมั่นคงปลอดภัย

ตารางที่ ค-4 รายการช่องโหว่ของ Moodle 4.5.1 ที่ตรวจพบจาก Nessus หลังการแก้ไขช่องโหว่

No.	Name	Level	Descrptiom
-	-	-	-
-	-	-	-

หลังการสแกน Moodle 4.5.1 หลังการแก้ไข ด้วย Nessus พบว่า ไม่มีช่องโหว่เหลืออยู่ ตามที่แสดงใน ตารางที่ ค-4 ซึ่งบ่งชี้ว่าการดำเนินการแก้ไขช่องโหว่เป็นไปอย่างมีประสิทธิภาพ อย่างไรก็ตาม เพื่อให้ระบบคงความมั่นคงปลอดภัยในระยะยาว ควรติดตามการอัปเดตซอฟต์แวร์และแพตช์ความมั่นคงปลอดภัยอย่างต่อเนื่อง รวมถึง ดำเนินการสแกนระบบเป็นระยะ เพื่อตรวจสอบและป้องกันช่องโหว่ใหม่ที่อาจเกิดขึ้นในอนาคต

ประวัติผู้เขียน

ชื่อ นายวุฒิภัทร พุ่มพวง

ชื่อการค้นคว้าอิสระ การประเมินช่องโหว่ของแพลตฟอร์มการจัดการเรียนการสอนออนไลน์

สาขาวิชา การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประวัติ สำเร็จการศึกษาระดับปริญญาตรี บริหารธุรกิจบัณฑิต สารสนเทศทางธุรกิจ มหาวิทยาลัยราชภัฏพระนครศรีอยุธยา

ประวัติการทำงาน

พ.ศ. 2554–2558	วิทยาลัยเทคโนโลยีพณิชยการอยุธยา ตำแหน่ง ช่างเทคนิคคอมพิวเตอร์
พ.ศ. 2558–2568	วิทยาลัยเทคโนโลยีพณิชยการอยุธยา ตำแหน่ง ครูผู้สอน แผนกคอมพิวเตอร์ธุรกิจและเทคโนโลยีสารสนเทศ
พ.ศ. 2564–2568	วิทยาลัยเทคโนโลยีพณิชยการอยุธยา ตำแหน่ง หัวหน้างานเทคโนโลยีสารสนเทศ
พ.ศ. 2568–ปัจจุบัน	วิทยาลัยเทคโนโลยีพณิชยการอยุธยา ตำแหน่ง รองผู้อำนวยการฝ่ายสารสนเทศและบริหารทรัพยากร