การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ

นายธีระพงษ์ จำปาทอง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ

นายธีระพงษ์ จำปาทอง

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิทยาศาสตรมหาบัณฑิต
สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ

โดย นายธีระพงษ์ จำปาทอง

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาการบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ

..... คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

(ผู้ช่วยศาสตราจารย์ ดร.สุนันฑา สดสี)

คณะกรรมการสอบการค้นคว้าอิสระ

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เทอดพงษ์ แดงสี)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์)

..... กรรมการ

(รองศาสตราจารย์ ว่าที่ร้อยตรี ดร.พงษ์พิสิฐ วุฒิดิษฐโชติ)

ชื่อ : นายธีระพงษ์ จำปาทอง
ชื่อการค้นคว้าอิสระ : การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชัน เพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ
สาขาวิชา : การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์
ปีการศึกษา : 2567

บทคัดย่อ

การค้นคว้าอิสระนี้มีวัตถุประสงค์เพื่อตรวจสอบความมั่นคงปลอดภัยระบบเครือข่ายและค้นหาช่องโหว่บนแอปพลิเคชันขององค์กร โดยการใช้เครื่องมือสแกนช่องโหว่ที่สามารถใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่าย เพื่อวิเคราะห์หาแนวทางการป้องกันช่องโหว่ที่เกิดจากการตั้งค่าระบบและช่องโหว่ที่อยู่บนเว็บแอปพลิเคชันขององค์กร และเพื่อดำเนินการลดความเสี่ยงที่จะเกิดขึ้นได้อย่างเหมาะสมและมีประสิทธิภาพ

ผู้วิจัยได้ทำการเลือกใช้เครื่องมือสแกนช่องโหว่จำนวน 3 เครื่องมือประกอบด้วย OpenVAS, Nessus, OWASP ZAP ซึ่งเป็นเครื่องมือสแกนช่องโหว่ที่ได้รับการยอมรับในปัจจุบัน โดยแต่ละเครื่องมือมีความสามารถที่แตกต่างกันในการตรวจสอบช่องโหว่ประเภทต่างๆ มาทดลองใช้ตรวจสอบช่องโหว่ในองค์กรซึ่งเป็นบริษัทเอกชนขนาดกลางแห่งหนึ่ง โดยจะทำการสแกน 2 ครั้ง ซึ่งการสแกนครั้งที่ 2 จะเป็นการสแกนเพื่อหาช่องโหว่ภายหลังจากที่ดำเนินการแก้ไขตามผลการสแกนครั้งที่ 1 และจะมีการเปรียบเทียบช่องโหว่ที่พบจากแต่ละเครื่องมือที่ใช้ประกอบกับการวิเคราะห์ความเสี่ยงที่จะเกิดขึ้น รายการช่องโหว่ที่ค้นพบจะถูกนำมาจัดกลุ่มรายการความเสี่ยงด้านความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน ตาม OWASP Top Ten 2021 และจัดทำเป็นรายงานการค้นพบช่องโหว่สำหรับนำเสนอต่อเจ้าหน้าที่ที่ดูแลเว็บแอปพลิเคชันขององค์กร เพื่อให้ผู้ดูแลระบบสามารถหาแนวทางการแก้ไขช่องโหว่ที่เกิดขึ้น ทำการปรับปรุงช่องโหว่พี่พบที่มีความเสี่ยงสูงหรือความเสี่ยงที่ต้องรีบทำการแก้ไข และจัดทำเป็นคู่มือการแก้ไขช่องโหว่ไว้ให้กับองค์กรได้ใช้ประโยชน์และใช้ในการพัฒนาด้านนโยบายความมั่นคงปลอดภัยไซเบอร์ภายในองค์กรอีกด้วย

(มีจำนวนทั้งสิ้น 77 หน้า)

คำสำคัญ : ช่องโหว่ เว็บแอปพลิเคชัน เครื่องมือสแกนช่องโหว่

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Theerapong Jampathong
Independent Study Title : Evaluating Web Application Vulnerability Scanning Tools for Developing Effective Attack Prevention Strategies.
Major Field : Network and Information Security Management
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Assistant Professor Nawaporn Wisitpongphan, Ph.D.
Academic Year : 2024

ABSTRACT

This independent study aims to assess network security and identify vulnerabilities in organizational applications using freely available vulnerability scanning tools. The objectives are to analyze preventive measures for vulnerabilities caused by system misconfigurations and web application weaknesses, and to effectively mitigate associated risks.

Three vulnerability scanning tools OpenVAS, Nessus, and OWASP ZAP were selected due to their widespread recognition and varying capabilities. These tools were employed in a case study conducted with a medium-sized private organization to assess their effectiveness. Two rounds of scanning were conducted, with the second scan performed after remediation of vulnerabilities identified in the first round. A comparative analysis of vulnerabilities detected by each tool was then conducted, along with a comprehensive risk analysis. The identified vulnerabilities were classified according to the OWASP Top Ten 2021 standards to create a detailed vulnerability report. This report was presented to the organization's web application administrators, providing clear insights for remediation. High-risk vulnerabilities or those requiring immediate attention were highlighted for prompt action. Additionally, a vulnerability remediation guide was created to assist the organization in ongoing cybersecurity policy development and implementation.

(Total 77 Pages)

Keywords: Vulnerability, Web Application, Vulnerability Scanning Tools

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความช่วยเหลือ และคำแนะนำให้คำปรึกษา ตลอดจนชี้แนะแนวทางที่เป็นประโยชน์ในการทำการค้นคว้าอิสระ และความกรุณาอย่างยิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.นวพร วิสิฐพงศ์พันธ์ อาจารย์ที่ปรึกษาการค้นคว้าอิสระนี้ ขอขอบคุณองค์กรที่ให้การสนับสนุนด้านทรัพยากรและข้อมูล และขอขอบคุณคณาจารย์ภาควิชาการบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศทุกท่านที่ได้ถ่ายทอดความรู้ในวิชาแขนงต่าง ๆ ซึ่งผู้วิจัยได้นำมาใช้ประโยชน์ในการจัดทำการค้นคว้าอิสระนี้

สุดท้ายนี้ผู้วิจัยขอกราบขอบพระคุณครอบครัว และเพื่อนๆ พี่น้องในภาควิชาการบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศที่ให้การสนับสนุน แลกเปลี่ยนความคิดเห็นตลอดเวลาระยะเวลาการทำวิจัย และเป็นกำลังใจให้เสมอมาซึ่งเป็นส่วนสำคัญต่อการค้นคว้าอิสระนี้สำเร็จลุล่วงไปได้ด้วยดี แม้ว่างานวิจัยครั้งนี้จะมีความท้าทายในหลายด้าน แต่กระบวนการศึกษาทำให้ได้รับประสบการณ์และความรู้ที่มีค่า ซึ่งจะเป็นพื้นฐานสำคัญในการพัฒนาและนำไปต่อยอดด้านความมั่นคงปลอดภัยทางไซเบอร์ในอนาคต

ธีระพงษ์ จำปาทอง

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูปภาพ

สารบัญรูปภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การโจมตีทางไซเบอร์ในปัจจุบันมีหลากหลายรูปแบบและเป็นภัยคุกคามต่อความมั่นคงปลอดภัยทางไซเบอร์เป็นอย่างมาก หน่วยงานต่างๆ ทั้งภาครัฐและเอกชนได้มีการจัดทำเว็บไซต์ของหน่วยงานตนเอง และมีการพัฒนาเพื่อให้บริการด้านต่างๆ แก่ประชาชน และทำให้เทคโนโลยีเกิดการพัฒนาขึ้นตลอดเวลา ผู้ไม่ประสงค์ดีมีการเรียนรู้เพื่อที่จะปรับตัวในการหาช่องทางที่จะใช้ประโยชน์จากเทคโนโนโลยีที่เกิดขึ้นอย่างต่อเนื่อง เพื่อสร้างช่องทางและโอกาสที่จะใช้ประโยชน์จากช่องโหว่ต่างๆ ที่เกิดขึ้น [1] ซึ่งถ้ายังไม่สามารถที่จะปิดช่องโหว่เหล่านี้อาจจะทำให้บริการเหล่านั้นเกิดความเสียหาย ข้อมูลสำคัญขององค์กรอาจจะตกอยู่ในมือของผู้ไม่หวังดีและอาจจะไม่สามารถดำเนินต่อไปได้ และทำให้องค์กรเสื่อมเสียชื่อเสียงได้

จากสถิติภัยคุกคามทางด้านไซเบอร์ จำแนกตามรูปแบบภัยคุกคาม ประจำปี พ.ศ. 2566 [2] ของสำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.) พบว่า รูปแบบภัยคุกคามที่มีจำนวนเกิดขึ้นมากที่สุด เกิดขึ้นจำนวน 515 เหตุการณ์ คือภัยคุกคาม Hacked Website (Gambling) การโจมตีเว็บไซต์ (เว็บพนันออนไลน์) และภัยคุกคามที่มีจำนวนเกิดขึ้นมากเป็นอันดับที่ 2 เกิดขึ้นจำนวน 336 เหตุการณ์ คือภัยคุกคาม Hacked Website (Defacement) การโจมตีเว็บไซต์เพื่อเปลี่ยนแปลงข้อมูลเผยแพร่บนหน้าเว็บไซต์ ซึ่งทั้งสองรูปแบบที่กล่าวมามีจำนวนเหตุการณ์เพิ่มขึ้นจากปีที่แล้วอีกด้วย และอีกทั้ง OWASP Top Ten 2021 [3] ยังมีการจัดอันดับความเสี่ยงด้านความมั่นคงปลอดภัยของเว็บแอฟพลิเคชัน ซึ่งทุก ๆ 4 ปี และมีการจัดอันดับช่องโหว่ขึ้นใหม่เพื่อให้องค์กรต่างๆ ทั่วโลก ได้นำข้อมูลนี้ไปใช้ปกป้องข้อมูลขององค์กร ได้อย่างถูกต้องและทันยุคทันสมัยต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น ได้แก่ 1. A01:2021-Broken Access Control, 2. A02-Cryptographic Failures, 3. A03-Injection, 4. A04-Insecure Design, 5. A05-Security Misconfiguration เป็นต้น ช่องโหว่ดังกล่าวที่ถูกประกาศขึ้นมานั้น บางช่องโหว่ยังคงต้องได้รับการป้องกันและตรวจสอบอย่างต่อเนื่อง เพื่อไม่ให้โดนโจมตีจากผู้ไม่หวังดีได้โดยง่าย ดังนั้น ในองค์กรจะต้องมีการทดสอบหรือทำการสแกนระบบเพื่อหาช่องโหว่ที่สามารถทำได้เองในองค์กร เพื่อให้รู้จุดอ่อนของเว็บแอปพลิเคชันในเบื้องต้นก่อนที่ผู้ไม่ประสงค์ดี จะใช้ประโยชน์จากช่องโหว่ขององค์กร [4] ด้วยเหตุนี้ผู้วิจัยจึงเล็งเห็นถึงความสำคัญในการป้องกันก่อนจะโดนผู้ไม่ประสงค์ดีโจมตีองค์กรจนทำให้เกิดความเสียหายแก่ทรัพย์สินและเสื่อมเสียชื่อเสียง ทุกองค์กรไม่ว่าจะขนาดใหญ่หรือเล็กควรมีวิธี

และแนวทางปฏิบัติในการรับมือจากการโดนใช้ประโยชน์จากช่องโหว่ ควรจัดให้มีการตรวจประเมินหรือวิเคราะห์ช่องโหว่เพื่อลดความเสี่ยงจากภัยคุกคามไซเบอร์ จากผู้ที่มีส่วนเกี่ยวข้องภายใน

องค์กรหรือผู้ให้บริการด้านความมั่นคงปลอดภัยไซเบอร์สำหรับหน่วยงานที่มีความต้องการไม่ให้ข้อมูลของภายในองค์กรรั่วไหลไปสู่ภายนอก การให้ความสำคัญกับความมั่นคงปลอดภัยของข้อมูลสามารถป้องกันได้โดยการหมั่นตรวจเช็คระบบต่างๆ [5] มีการทำรายงานผลช่องโหว่ที่พบ และหาวิธีการแนวทางการแก้ไขปิดช่องโหว่ดังกล่าวให้ได้ เพื่อไม่ให้องค์กรตกเป็นเหยื่อของผู้ไม่ประสงค์ดี

1.2 วัตถุประสงค์

1.2.1 เพื่อเปรียบเทียบซอฟต์แวร์ที่ใช้สำหรับตรวจสอบช่องโหว่ที่เหมาะสมสำหรับองค์กรเอกชนขนาดกลาง

1.2.2 เพื่อตรวจสอบและวิเคราะห์ช่องโหว่ที่เกิดจากการโจมตีการใช้ประโยชน์ จากข้อผิดพลาดในการตั้งค่าระบบหรือช่องโหว่ของเว็บแอปพลิเคชันขององค์กร

1.2.3 เพื่อจัดทำรายงานช่องโหว่ที่เกิดขึ้นส่งให้ผู้ดูแลเว็บแอปพลิเคชันขององค์กรได้

1.2.4 เพื่อหาแนวทางการป้องกันและแก้ไขช่องโหว่หรือจุดอ่อนที่จะเป็นช่องทางการโจมตีในอนาคตสำหรับเว็บแอปพลิเคชันขององค์กร

1.3 ขอบเขตการวิจัย

1.3.1 ขอบเขตของการตรวจสอบช่องโหว่

1.3.1.1 เว็บไซต์ขององค์กรของเอกชนแห่งหนึ่ง เป็นระบบประเภทเว็บไซต์การเข้าใช้งานทั่วไป มีการใช้งานในระบบมากกว่า 9,000 ครั้งต่อเดือน

1.3.1.2 ช่องโหว่ที่ตรวจสอบเป็นช่องโหว่ที่ระบุอยู่ใน OWASP Top Ten 2021

1.3.2 ขอบเขตด้านซอฟต์แวร์ที่ใช้สำหรับตรวจสอบช่องโหว่

1.3.2.1 OpenVAS (Open Vulnerability Assessment System)

1.3.2.2 Nessus vulnerability scanner

1.3.2.3 OWASP ZAP

1.4 คำจำกัดความในงานวิจัย

1.4.1 ช่องโหว่ (Vulnerability) เป็นข้อบกพร่องหรือข้อผิดพลาดของโค้ด ส่วนมากจะพบในเว็บไซต์ ซึ่งเกิดจากระบบหลังบ้านจากการเขียนโค้ดที่ไม่ละเอียดรอบคอบ ซึ่งเมื่อไม่มีการแก้ไขหรืออัปเดตอยู่เป็นประจำจะทำให้ ผู้ที่ไม่ประสงค์ดีเข้าถึงระบบหรือข้อมูลโดยไม่ได้รับอนุญาต

1.4.2 เว็บแอปพลิเคชัน (Web Application) เป็นโปรแกรมประยุกต์ที่สามารถเข้าถึงเพื่อใช้งานผ่านเว็บเบราว์เซอร์ ปัจจุบันยังได้รับความนิยมเป็นอย่างมาก เพราะสามารถใช้งานได้ง่าย และมีการติดตั้งที่ไม่ซับซ้อน

1.4.3 ความเสี่ยง (Risk) คือโอกาสที่จะเกิดเหตุการณ์ไม่พึงประสงค์ซึ่งอาจส่งผลกระทบต่อการทำงานของระบบ ความมั่นคงปลอดภัยของข้อมูล หรือทรัพย์สินขององค์กร

1.4.4 เครื่องมือสแกนช่องโหว่ (Vulnerability Scanning Tools) โปรแกรมหรือซอฟต์แวร์ที่ใช้ในการตรวจสอบและค้นหาช่องโหว่ด้านความมั่นคงปลอดภัยในระบบคอมพิวเตอร์ เครือข่าย หรือแอปพลิเคชัน โดยจะทำการทดสอบและรายงานจุดอ่อนที่ถูกตรวจพบซึ่งอาจถูกโจมตีได้ในอนคต

1.4.5 การตรวจสอบช่องโหว่ทางเทคนิค (Vulnerability Assessment) [7] เป็นกระบวนการตรวจสอบระบุ วิเคราะห์ และจัดลำดับความสำคัญของช่องโหว่ด้านความมั่นคงปลอดภัยในระบบการตั้งค่าที่ไม่ถูกต้อง จนทำให้มีความเสี่ยงด้านความมั่นคงปลอดภัยมีจุดอ่อนที่อาจจะโดนโจมตีจากผู้ไม่ประสงค์ดีได้ เพื่อตรวจสอบความมั่นคงปลอดภัย จำเป็นต้องมีขั้นตอนตั้งแต่การวางแผน การสแกนหาช่องโหว่ การวิเคราะห์ผล และการจัดทำรายงานพร้อมข้อเสนอแนะในการแก้ไข

1.5 ประโยชน์ของการวิจัย

1.5.1 องค์กรรู้ถึงข้อผิดพลาดในการตั้งค่าระบบ และช่องโหว่ของเว็บแอปพลิเคชันองค์กร

1.5.2 องค์กรมีเครื่องมือและแนวทางการป้องกันแก้ไขช่องโหว่ที่เกิดขึ้น

1.5.2 องค์กรสามารถวางแผนเพื่อดำเนินการลดความเสี่ยงที่จะเกิดขึ้นได้อย่างเหมาะสมและมีประสิทธิภาพ

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การค้นคว้าอิสระเรื่อง การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ เป็นการศึกษาหาแนวทางแก้ช่องโหว่บนเว็บแอปพลิเคชันขององค์กร โดยการนำเครื่องมือสแกนช่องโหว่ที่ไม่มีค่าใช้จ่ายมาใช้ให้เกิดประโยชน์ภายในองค์กรได้อย่างมีประสิทธิภาพ ผู้วิจัยจึงได้ศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางในการค้นคว้าดังต่อไปนี้

2.1 ทฤษฎีเกี่ยวกับความมั่นคงปลอดภัยทางไซเบอร์

2.2 ทฤษฎีเกี่ยวกับการประเมินช่องโหว่

2.3 OWASP TOP 10 2021

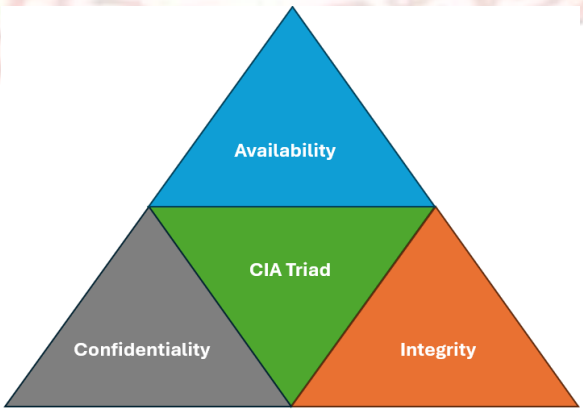
2.4 เครื่องมือสำหรับการสแกนตรวจหาช่องโหว่

2.5 งานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีเกี่ยวกับความมั่นคงปลอดภัยทางไซเบอร์

2.1.1 CIA Triad (Confidentiality, Integrity, Availability)

ปัจจุบันความมั่นคงปลอดภัยทางด้านไซเบอร์ (Cybersecurity) เป็นเรื่องที่สำคัญและใกล้ตัวมาก เนื่องจากข้อมูลสำคัญหรือทรัพย์สินภายในองค์กรเป็นเรื่องที่สำคัญ และมีมูลค่าทั้งสิน จึงควรมีวิธีการปกป้องข้อมูลให้ปลอดภัย โดยอาศัยองค์ประกอบด้านความมั่นคงปลอดภัยทางไซเบอร์พื้นฐาน 3 ข้อ เพื่อให้ข้อมูลมีความมั่นคงปลอดภัยและพร้อมใช้งานตลอดเวลา [6] ดังนี้



ภาพที่ 2-1 องค์ประกอบด้านความมั่นคงปลอดภัยทางไซเบอร์ CIA Triad

2.1.1.1 Confidentiality คือ การป้องกันข้อมูลหรือทรัพยากรที่มีความสำคัญไม่ให้ถูกเปิดเผยหรือเข้าถึงโดยบุคคลที่ไม่ได้รับสิทธิ์และ รักษาความเป็นส่วนตัว จำกัดสิทธิ์การเข้าถึงข้อมูล กำหนดระดับชั้นความลับของข้อมูล อนุญาตเฉพาะบุคลที่มีสิทธิ์ในการเข้าถึงข้อมูล

2.1.1.2 Integrity คือ การป้องกันไม่ให้ข้อมูลหรือระบบต่างๆถูกเข้าไปดัดแปลงหรือแก้ไขโดยไม่ได้รับอนุญาต เพื่อให้ข้อมูลมีความถูกต้อง ความน่าเชื่อถือ และสมบูรณ์อยู่เสมอ เช่น การตรวจสอบความถูกต้องของข้อมูล การสำรองข้อมูล

2.1.1.3 Availability คือ ความสามารถในการที่จะนำข้อมูลมาใช้ การทำให้ระบบสารสนเทศหรือข้อมูลสารสนเทศสามารถเข้าถึงข้อมูลหรือระบบได้อย่างต่อเนื่องมีประสิทธิภาพสูงสุดและ ระบบสามารถให้บริการได้ตามความต้องการของผู้ใช้งาน

2.1.2 NIST Cybersecurity Framework

NIST Cybersecurity Framework เป็นกรอบการทำงานด้านความมั่นคงปลอดภัยไซเบอร์ที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน หลายหน่วยงานได้เริ่มมีการนำ Framework นี้มาประยุกต์ใช้ในหน่วยงานเพื่อรับมือกับภัยคุกคามทางไซเบอร์ เนื่องจากมีหลักการและแนวทางปฏิบัติที่ดีด้านการบริหารจัดการความเสี่ยงและการจัดการกับภัยคุกคามที่จะเกิดขึ้นได้อย่างมีประสิทธิภาพ เพื่อยกระดับความมั่นคงปลอดภัยของหน่วยงานหรือองค์กรในทุกระดับ และยังช่วยให้หน่วยงานหรือองค์กรสามารถวางแผนรับมือป้องกัน ตรวจจับ และตอบสนองต่อภัยคุกคามได้อย่างทันท่วงที ในขณะที่ธุรกิจยังคงสามารถดำเนินการได้อย่างต่อเนื่อง นอกจากนี้ยังช่วยให้หน่วยงานลดความเสี่ยงในการถูกโจมตีจากผู้ไม่หวังดีได้ โดยจะสามารถแบ่งประเด็นการจัดการออกเป็น 6 ห้วข้อหลัก[7] คือ

ภาพที่ 2-2 กรอบการทำงานด้านความมั่นคงปลอดภัยไซเบอร์ NIST Cybersecurity Framework

2.1.2.1 Govern (การกำกับดูแล) มีกรอบนโยบายควบคุมจากการพิจารณาความเสี่ยงที่จะเกิดขึ้นกับองค์กร ผลกระทบที่จะเกิดขึ้นจากภัยทางไซเบอร์และมีเป้าหมายของการรักษาความมั่นคงปลอดภัย

2.1.2.2 Identify (การระบุ) การระบุทรัพย์สินและมีความเข้าใจถึงบริบทต่างๆในหน่วยงานเพื่อบริหารจัดการความเสี่ยง

2.1.2.3 Protect (การป้องกัน) มีมาตรฐานเพื่อป้องกันข้อมูลหรือระบบของหน่วยงานไม่ให้เกิดความเสียหาย

2.1.2.4 Detect (การตรวจจับ) การกำหนดวิธีการขั้นตอนและกระบวนการต่างๆ เพื่อตรวจจับเหตุการณ์ที่ไม่ปกติ

2.1.2.5 Respond (การตอบสนองต่อภัยคุกคาม) การกำหนดวิธีการขั้นตอนและกระบวนการต่างๆ เพื่อรับมือกับเหตุการณ์ที่ไม่ปกติที่เกิดขึ้น

2.1.2.6 Recovery (การกู้คืนระบบ) การกำหนดวิธีการขั้นตอนและกระบวนการต่างๆ เพื่อให้ธุรกิจสามารถดำเนินการไปได้อย่างต่อเนื่อง และสามารถฟื้นฟูระบบให้กลับมาใช้งานได้อย่างปกติ

2.2 ทฤษฎีเกี่ยวกับการประเมินช่องโหว่

Vulnerability Assessment (การประเมินช่องโหว่) [8] คือขั้นตอนที่ใช้ทดสอบว่าเว็บแอปพลิชันขององค์กรมีช่องโหว่มากน้อยเพียงใด ระดับความรุนแรงสูงขนาดไหน มีความมั่นคงปลอดภัยหรือไม่ ซึ่งในการสแกนทดสอบนั้นเพื่อค้นหาช่องโหว่สำคัญ ข้อมูลที่อาจถูกเปิดเผย โดยมีเป้าหมายคือเว็บแอปพลิเคชันขององค์กร

ระดับความรุนแรงของช่องโหว่โดยทั่วไปจากเครื่องมือสแกน (เช่น OpenVAS, Nessus) โดยระดับความรุนแรงของ CVSS v3.0 (Common Vulnerability Scoring System) ใช้เกณฑ์การให้คะแนนได้ 4 ระดับคือ

ระดับความรุนแรงวิกฤต (Critical) คะแนน: CVSS 9.0-10 มีผลกระทบร้ายแรง ควรแก้ไขทันที ลักษณะคือช่องโหว่ที่ทำให้ผู้ไม่ประสงค์ดีสามารถเข้าถึงระบบได้อย่างสมบูรณ์ (Full System-Compromise) การโจมตีสามารถทำได้ง่าย และไม่จำเป็นต้องมีสิทธิ์การใช้งานในระบบก่อน ตัวอย่างช่องโหว่ที่พบทั่วไป เช่น Remote Code Execution (RCE) ผู้โจมตีสามารถส่งคำสั่งอันตรายจากระยะไกลมารันบนระบบได้ และ SQL Injection (SQLi) ไม่ประสงค์ดีสามารถส่งคำสั่ง SQL ผ่านช่องทางการกรอกข้อมูลของเว็บและเข้าถึงฐานข้อมูลได้

ระดับความรุนแรงสูง (High) คะแนน: CVSS 7.0 – 8.9 มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย ลักษณะคือ ช่องโหว่ที่อาจทำให้ผู้โจมตีเข้าถึงระบบได้บางส่วน หรือทำให้เกิดความเสียหายร้ายแรง แต่ยังไม่ถึงระดับควบคุมทั้งระบบ ต้องมีเงื่อนไงเพิ่มเติม เช่น ผู้ใช้ทำการคลิกลิงก์ที่เป็นอันตราย ตัวอย่างช่องโหว่ที่พบทั่วไป เช่น Cross-Site Scripting (XSS) ผู้ไม่ประสงค์ดีสามารถฝังโค้ด JavaScript อันตรายผ่านช่องทางการกรอกข้อมูลของเว็บไซต์ ทำให้ขโมย Cookie หรือข้อมูลสำคัญได้ และ Privilege Escalation ช่องโหว่ที่ทำให้ผู้ใช้งานทั่วไปสามารถยกระดับสิทธิ์กลายเป็น Administrator ได้

ระดับความรุนแรงปานกลาง (Medium) คะแนน: CVSS 4.0 - 6.9 มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี ลักษณะคือ เป็นช่องโหว่ที่สร้างผลกระทบต่อระบบ แต่ไม่ทำให้ระบบถูกยึดครองระบบได้โดยตรง ต้องมีปัจจัยหลายอย่างร่วมกันเพื่อทำการโจมตีให้สำเร็จ ตัวอย่างช่องโหว่ที่พบทั่วไป เช่น การเปิดเผยข้อมูลจากไฟล์ที่เข้าถึงได้ง่าย หรือหน้าที่มีการแสดงข้อมูลภายใน Disclosure และ การใช้งานซอฟต์แวร์ที่ล้าสมัย มีช่องโหว่ที่อาจถูกใช้โจมตี แต่ยังไม่ถึงขั้นควบคุมระบบโดยตรงทันที

ระดับความรุนแรงต่ำ (Low) คะแนน: CVSS: 0.0-3.9 มีผลกระทบต่ำ มีความมั่นคงปลอดภัยไม่สามารถโจมตีได้ง่าย ลักษณะคือ เป็นช่องโหว่ที่มีผลกระทบน้อยมาก ไม่สามารถโจมตีระบบได้โดยตรง ใช้การโจมตีร่วมกับช่องโหว่อื่น ส่งผลกระทบน้อย ตัวอย่างช่องโหว่ที่พบทั่วไป เช่น เว็บเซิร์ฟเวอร์เปิดเผยเวอร์ชันซอฟต์แวร์ใน HTTP Header และ ขาดการตั้งค่า Security Header บางรายการที่จำเป็น

2.3 OWASP TOP 10 2021

OWASP หรือ Open Web Application Security Project คือ มาตราฐานความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน มีการจัดอันดับช่องโหว่ที่มีความรุนแรง จัดทำขึ้นโดยองค์กรที่ไม่แสวงหาผลกำไรที่ให้ความรู้และเป็นเครื่องมือสำหรับการพัฒนาเว็บแอปพลิเคชันให้มีความมั่นคงปลอดภัย นักพัฒนาเว็บแอปพลิเคชันทั่วโลกให้การยอมรับ และเป็นมาตรฐานการทดสอบช่องโหว่ เว็บแอปพลิเคชันที่อธิบายรายละเอียดของช่องโหว่ที่พบและมีระดับความรุนแรง 10 อันดับแรกในปี 2021 [9] มีดังนี้

ภาพที่ 2-3 มาตรฐานความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน OWASP TOP 10 2021

2.3.1 A01-Broken Access Control

เกิดจากสิทธิ์ในการใช้งานใน user ทำได้มากเกินกว่าที่ user level นั้นควรจะทำได้ คือ การควบคุมสิทธิ์การเข้าถึงที่ไม่รัดกุม ทำให้ผู้ใช้ที่ไม่ได้รับอนุญาตสามารถเข้าถึงข้อมูลหรือฟังก์ชันที่ควรถูกจำกัดได้ ซึ่งอาจนำไปสู่การขโมยข้อมูล แก้ไขข้อมูล หรือเข้าควบคุมระบบโดยไม่ได้รับอนุญาต ดังนั้นต้องมีการจำกัดสิทธิ์การเข้าถึง หรือกำหนดสิทธิ์อย่างเหมาะสมกับการใช้งานและตรวจสอบสิทธิ์ในทุกคำขอใช้งานล้าสมัยแล้ว

2.3.2 A02-Cryptographic Failures

การเข้าถึงข้อมูลส่วนตัว หรือข้อมูลที่เป็น database โดยไม่ได้ทำการเข้ารหัสข้อมูลเหล่านี้ไว้ ความล้มเหลวในการเข้ารหัสข้อมูลสำคัญ จนนำไปสู่การถูกผู้ไม่ประสงค์ดีสามารถนำข้อมูลออกจากฐานข้อมูลได้ ซึ่งทำให้มีความเสียหายสูง หากข้อมูลอยู่ในมือของผู้ไม่ประสงค์ดีอาจจะโดนนำข้อมูลไปหลอก เป็น Call Center ว่าเป็นเจ้าของข้อมูลตัวจริงได้ ซึ่งจะส่งผลกระทบร้ายแรงต่อเจ้าของตัวจริง ได้ช่องโหว่นี้รวมถึงการไม่ได้เก็บข้อมูลรหัสผ่านแบบ Hash หรือใช้ Password Hashing Function เช่น ไม่เข้ารหัสข้อมูลส่วนบุคคล หรือใช้วิธีการเข้ารหัสที่ไม่ปลอดภัย ดังนั้น ต้องมีการเข้ารหัสข้อมูลสำคัญด้วยวิธีที่ปลอดภัยและทันสมัย

2.3.3 A03-Injection

เป็นหนึ่งในช่องโหว่ด้านความมั่นคงปลอดภัยที่อันตรายมาก โดยเกิดขึ้นเมื่อแอปพลิเคชัน รับข้อมูลจากผู้ใช้โดยไม่มีการตรวจสอบและกรองอย่างเหมาะสม ทำให้แฮ็กเกอร์สามารถแอบแทรกคำสั่งอันตรายไปยังฐานข้อมูล ระบบปฏิบัติการ หรือส่วนอื่นๆ ของแอปพลิเคชันได้โดยตรง จะส่งผลกระทบอย่างร้ายแรงต่อฐานข้อมูลของ User คนอื่นๆ ช่องโหว่ยอดนิยมมากที่สุดคือ SQL Injection และ Injection ผ่าน XSS และยังมีรูปแบบอื่นๆอีกมากมาย เช่น NoSQL Injection, System Call Injection / OS Command Injection, LDAP Injection, Expression Language- Injection, Path Manipulation เป็นต้น

2.3.4 A04-Insecure Design

การออกแบบระบบโครงสร้างที่ผิดพลาดทำให้ไม่ปลอดภัย ซึ่งจะไม่ใช่ปัญหาที่เกิดจากการดำเนินการที่ไม่ถูกวิธี ปัญหาเกี่ยวกับการตั้งค่าที่ไม่ปลอดภัย ทำให้ถูกโจมตีได้โดยง่าย หรือ การออกนโยบายที่ผิดพลาด เช่น การเก็บข้อมูลที่เป็น Key ต่างๆ ไว้ที่ Client ควรเก็บไว้บน Server การอนุญาตให้แก้ไขข้อมูลสำคัญเช่น email ควรถาม password ก่อน ถ้าไม่อย่างนั้นจะทำให้บุคคลอื่นสามารถกด forgot password เพื่อเปลี่ยนแลงรหัสผ่านใหม่ได้ทันทีทำให้ไม่สามารถที่จะเข้า email ได้ และยังรวมไปถึง ซอฟต์แวร์ล้าสมัยและไม่ได้รับการอัปเดตอย่างสม่ำเสมอ

2.3.5 A05-Security Misconfiguration

การตั้งค่าระบบที่ไม่ถูกต้อง หรือลืมตั้งค่า ส่งผลทำให้มีช่องโหว่ที่ไม่ได้คิดไว้หลุดออกไป ทำให้ระบบเปิดโอกาสให้กับผู้ไม่ประสงค์ดีเข้ามาได้ เช่น การใช้ค่า default ของซอฟต์แวร์ต่างๆ ควรมีการแก้ไข username และ password ที่มาจากโรงงาน เพื่อไม่ให้เดารหัสผ่านได้โดยง่าย เปิดหน้า Error Message ออกแสดงบนหน้า Web Browser ทำให้รู้ข้อมูลสำคัญ เปิด port ที่ไม่จำเป็น

2.3.6 A06-Vulnerable and Outdated Components

ช่องโหว่นี้เป็นช่องโหว่ยอดนิยมสำหรับองค์กรที่มี Software ที่มีการใช้งานมายาวนานใช้มาต่อเนื่องหลายปี และไม่กล้า Upgrade software ที่ใช้เพราะกลัวว่า Software ที่ใช้งานมานานจะทำงานไม่ได้ เช่น หลายองค์กรยังคงใช้ Internet Explorer หรือ Windows XP ทั้งๆ ที่รู้ว่าหมดระยะเวลา Support Security patch มานานแล้ว Feature ใหม่ๆ ก็ไม่มี แต่ยังอดทนใช้ต่อ จึงเป็นความเสี่ยงที่เหมือนองค์กรรู้อยู่ สุดท้ายหากเกิดโดนผู้ไม่หวังดี hack ขึ้นมาก็อาจมีการสูญเสียเกิดขึ้น วิธีป้องกันคือต้องมีการ upgrade software ขององค์กรอย่างต่อเนื่องหรือยอมลงทุนสร้างใหม่ทั้งหมด

2.3.7 A07-Identification and Authentication Failures

ช่องโหว่การทำงานของส่วนที่เกี่ยวข้องกับการเข้าถึงระบบหรือช่องทางตรวจสอบยืนยันตัวตน สามารถขโมย Identity ของผู้ใช้หรือเดาสุ่ม password จนถูกและเข้ายึด account ได้ ซึ่งเกิดได้จากหลายสาเหตุ เช่น การใช้ Password ที่ไม่เข้มแข็งหรือเดาได้ง่าย มีการทำ Brute Force username, password ได้, ใช้ default password เหมือนกันทุก user, ใช้กระบวนการ forgot password ไม่มีประสิทธิภาพ ไม่ทำการ Hash Password, access token hi-jack จากการไม่ใช้ https ในการรับส่งข้อมูล ดังนั้นจึงควรป้องกันโดยการ ใช้ Multi Factor Authentication, ไม่ส่ง default password ที่เหมือนกันทุก user, เพิ่มระบบ Weak password check, จำกัดการ Login ไม่ให้ผิดพลาดตามจำครั้งที่ได้จำกัดไว้

2.3.8 A08-Software and Data Integrity Failures

ส่วนนี้คือช่องโหว่ของการติดตั้ง software หรือการถูกสอดแทรก code บางอย่างให้มารันตอน integration โดยไม่ได้ตั้งใจ เช่น การ upgrade software โดยไม่ตรวจสอบถูกต้องหรือความเหมาะสมความเข้ากันได้ของ software, การทำ CI/CD pipelines ไม่ได้มีการทดสอบการทำงานร่วมกัน หรือปล่อยให้บุคคลภายนอกที่ไม่มีความเกี่ยวข้องสามารถเข้าถึงได้, ไม่ได้มีการตรวจสอบ Digital Signature วันเวลาลง Software ใดๆ ใครเป็นคนติดตั้ง, ผู้โจมตีสามารถรันโค้ดบนเครื่องเป้าหมายผ่านเครือข่ายอินเทอร์เน็ตหรือเครือข่ายภายในได้ โดยที่ไม่ได้รับอนุญาต

2.3.9 A09-Security Logging and Monitoring Failures

ความล้มเหลวในการบันทึกและตรวจสอบเหตุการณ์ด้านความมั่นคงปลอดภัย ทำให้ไม่สามารถตรวจจับและตอบสนองต่อภัยคุกคามหรือการโจมตีได้อย่างทันท่วงที โดยทั่วไปเมื่อเกิดเหตุการณ์ไปแล้ว และมักจะพบภายหลัง โดยบริษัทภายนอกเสมอ ซึ่ง developer อาจไม่ได้ Log พฤติกรรมสำคัญๆให้ครบ หรือ Application ไม่สามารถแจ้งเตือนเมื่อพบเหตุการณ์ผิดปกติได้ตลอดเวลา ดังนั้นจึงควรจะมีการตรวจจับพฤติกรรมสำคัญๆ ให้ครบถ้วนมากที่สุด และเช็คให้แน่ใจว่า Log ถูก generate ใน format ที่ centralized log management เข้ามาเก็บข้อมูลไปใช้ได้ง่ายเพื่อให้สามารถรวม Log มาอ่านได้ง่าย มีระบบแจ้งเตือนเมื่อมีเหตุการณ์ที่ผิดปกติเกิดขึ้น จะได้รู้ก่อนและหาทางแก้ไขให้เร็วที่สุด

2.3.10 A10-Server-Side Request Forgery

เกี่ยวข้องกับการควบคุมเซิร์ฟเวอร์ ซึ่งผู้ไม่หวังดีสามารถทำให้เซิร์ฟเวอร์ไปร้องขอบริการจากเครื่องอื่นในเครือข่ายโดยการเปลี่ยนโปรโตคอล URL หรือพอร์ตในช่องทางที่อนุญาตให้ใส่ URL แต่แอปพลิเคชันไม่ได้ตรวจสอบข้อมูลที่รับเข้ามาด้วยว่าถูกต้องหรือไม่ การปลอม request ฝั่ง server ซึ่งเกิดจากการอนุญาตให้มีการ fetch ข้อมูลจากฝั่ง server เองจากภายนอกมาใช้งาน แล้วมีการใช้ script ในฝั่ง Server ยิงไปตาม parameter URL ส่งผลให้สามารถเข้าถึงข้อมูลไฟล์ที่สำคัญบน server ได้รวมถึงฐานข้อมูล ทั้งที่บุคคลที่ไม่เกี่ยวข้องไม่ควรจะเข้าถึง

2.4 เครื่องมือสำหรับการสแกนตรวจหาช่องโหว่

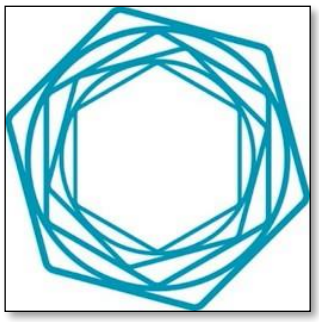
เครื่องมือสำหรับการสแกนตรวจหาช่องโหว่ นำมาใช้เพื่อทดสอบในการตรวจหาช่องโหว่ในเว็บแอปพลิเคชันสามารถทดสอบความมั่นคงปลอดภัยเพื่อป้องกันการโจมตี จากผู้ไม่ประสงค์ดีได้ สำหรับการค้นคว้าอิสระนี้เลือกใช้โปรแกรมที่มีความนิยมสูง ใช้งานง่าย ได้ผลลัพท์ที่ครอบคลุม ได้แก่ OpenVAS, Nessus และ OWASP ZAP

2.4.1 OpenVAS (Open Vulnerability Assessment System)

ภาพที่ 2-4 สัญลักษณ์ OpenVAS

OpenVAS เป็นเครื่องมือที่ใช้สำหรับทำการสแกนเพื่อประเมินช่องโหว่แอปพลิเคชัน [10] ที่สามารถใช้งานได้ฟรีโดยไม่มีค่าใช้จ่าย เป็นที่นิยมที่ใช้ในการทดสอบสแกน และสามารถตรวจสอบระบบเครือข่ายเพื่อดูว่ามีช่องโหว่ที่ถูกเปิดเผย หรือ มีความเสี่ยงต่อความมั่นคงปลอดภัยหรือไม่ มีการอัปเดตภัยคุกคามอยู่สม่ำเสมอ ซึ่งสามารถวิเคราะห์และประเมินระดับความรุนแรงของช่องโหว่ได้ มีผลลัพท์ที่ครอบคลุม ข้อมูลน่าเชื่อถือ ใช้งานง่าย มีคำแนะนำในการแก้ไขช่องโหว่ต่างๆเพื่อที่จะลดความเสี่ยงที่จะเกิดขึ้นลงได้

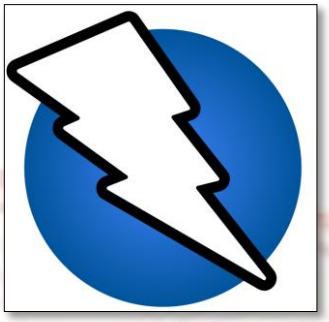
2.4.2 Nessus vulnerability scanner



ภาพที่ 2-5 สัญลักษณ์ Nessus

Nessus vulnerability scanner เป็นเครื่องมือที่สามารถใช้งานได้ฟรีโดยไม่มีค่าใช้จ่าย [11] มีคุณสมบัติในการค้นหาอุปกรณ์ที่อยู่ในระบบเครือข่าย และระบุว่าเป็นประเภทใด มีการใช้งานบริการอะไรอยู่บ้าง การเปิดใช้งานพอร์ตอะไรบ้าง บอกระดับความเสี่ยง มีช่องโหว่ที่เสี่ยงต่อการโจมตี โดยอ้างอิงจากมาตรฐานสากล เช่น CVE (Common Vulnerabilities and Exposures) มีวิธีการป้องกันการแก้ไขช่องโหว่ มีความสามารถในการสแกนได้รวดเร็วมีความแม่นยำ เวลาในการสแกนน้อย และสามารถตรวจจับช่องโหว่ได้อย่างมีประสิทธิภาพ

2.4.3 OWASP ZAP



ภาพที่ 2-6 สัญลักษณ์ OWASP ZAP

OWASP ZAP เป็นเครื่องมือที่สามารถใช้งานได้ฟรีโดยไม่มีค่าใช้จ่าย สามารถทำการทดสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชันได้อย่างมีประสิทธิภาพ สามารถค้นหาช่องโหว่หรือจุดอ่อนที่อาจทำให้ผู้ไม่ประสงค์ดีเข้าถึงข้อมูลที่สำคัญ มีความยืดหยุ่นและมีคุณสมบัติที่หลากหลาย ขยายขีดความสามารถได้ และสามารถทำงานได้บนหลายระบบปฏิบัติการ เช่น Linux Windows และอื่นๆ โดย OWASP ZAP เป็นเครื่องมือที่ใช้ในการสแกนหาช่องโหว่ที่ใช้กันแพร่หลาย [12]

2.5 งานวิจัยที่เกี่ยวข้อง

จากการทบทวนงานวิจัยที่เกี่ยวกับการตรวจสอบช่องโหว่ในองค์กร พบว่ามีงานวิจัยที่เกี่ยวข้อง 3 ประเภท คืองานวิจัยที่มุ่งเน้นการนำเครื่องมือสแกนช่องโหว่มาเปรียบเทียบและประยุกต์ใช้ในองค์กร งานวิจัยที่มุ่งเน้นการพัฒนาเครื่องมือสำหรับตรวจสอบช่องโหว่ และงานวิจัยที่มีการกำหนดกรอบและวิธีการตรวจสอบช่องโหว่ในองค์กร

จากการศึกษาพบว่า งานวิจัยที่เน้นการเปรียบเทียบประสิทธิภาพของเครื่องมือสแกนช่องโหว่นั้นนิยมเปรียบเทียบเครื่องมือแบบ Open Source ประกอบด้วย OpenVAS, Nessus และ OWASP ZAP โดยส่วนมากมักนำเครื่องมือมาทดลองสแกนหาช่องโหว่ในองค์กรของตนเอง ตัวอย่างงานวิจัยที่เน้นการเปรียบเทียบประสิทธิภาพของเครื่องมือ มีดังนี้

รุ้งตะวัน ประสนิท และ คณะ [13] เลือกใช้เครื่องมือ OpenVAS ในการจัดการและ ความคุมกระบวนการสแกนช่องโหว่ซึ่งเป็นหนึ่งในกระบวนการที่มีความสำคัญต่อผู้ดูแลระบบ และการรักษาความมั่นคงปลอดภัยสารสนเทศขององค์กร โดยได้ทำการทดสอบตรวจสอบช่องโหวทางเทคนิคของระบบปฏิบัติการและแอปพลิเคชันของ บริษัทแห่งหนึ่ง รวมถึงศึกษาความพึงพอใจของผู้ใช้งานระบบจากกลุ่มตัวอย่างที่ใช้ คือกลุ่มผู้ดูแลระบบ จำนวน 3 คน ซึ่งได้มาด้วยวิธีการแบบเจาะจง

ความพึงพอใจด้านมาตรฐานเนื้อหาและด้านการประยุกต์แนวทางการตรวจสอบระบบเทคโนโลยีสารสนเทศเทียบกับมาตรฐาน ISO/IEC 27001:2022 มีคุณภาพอยู่ในระดับดี

Aksu และ คณะ [10] ประเมินเครื่องมือ OpenVAS 9.0 ที่ใช้ในการสแกนหาช่องโหว่โดยใช้ผู้เชี่ยวชาญ และผู้ใช้งานทั่วไปที่มีความรู้ด้านความมั่งคงปลอดภัย เพื่อหาข้อบกพร่องด้านความสามารถในการใช้งานในสถานการณ์จริง ซึ่งในการทดสอบการใช้งานนั้นมีการกำหนดค่าตั้งต้นในการสแกนที่เหมาะสม และระบุให้ผู้ทดสอบดำเนินการตรวจจับช่องโหว่ให้ได้มากที่สุด เมื่อเปรียบเทียบแล้วพบว่าเครื่องมือ OpenVAS นั้นยังมีจุดบกพร่องในส่วนการของการนำเสนอข้อมูลบน Task Wizard ที่ช่วยตั้งค่าการตรวจสอบช่องโหว่ ซึ่งส่งผลให้ผู้ใช้งานทั่วไปถึง 70% ไม่สามารถใช้เครื่องมือได้อย่างเต็มประสิทธิภาพ

นอกจากนี้ยังมีการเปรียบเทียบประสิทธิภาพการทำงานของเครื่องมือที่ใช้ในการสแกนหาช่องโหว่แบบใช้งานฟรี ที่เป็นที่นิยมตัวอื่น เช่น Nessus และ Retina [11] โดยสิ่งที่ใช้เปรียบเทียบคือระยะเวลาในการสแกน และความสามารถในการตรวจจับช่องโหว่ ซึ่งผลการทดสอบพบว่าเครื่องมือทั้งสองสามารถตรวจจับช่องโหว่ได้ผลลัพธ์เป็นที่น่าพอใจ แต่Nessus มีความละเอียดและแม่นยำมากกว่าจึงทำให้เครื่องมือ Nessus มีข้อได้เปรียบเล็กน้อยจากการสแกนนี้

ในด้านการพัฒนาเครื่องมือสำหรับตรวจจับช่องโหว่นั้น มีกลุ่มนักวิจัยหลายกลุ่มที่เน้นการศึกษาและพัฒนาเครื่องมือทั้งที่เป็นมาตรฐาน และเครื่องมือที่พัฒนาขึ้นมาใหม่เพื่อการตรวจจับช่องโหว่ ตัวอย่างเช่น

ทัญญะ สิทธิมณีวรรณ [14] พัฒนากรอบการจัดการช่องโหว่ระบบสารสนเทศขององค์กร โดยได้นำขั้นตอนจากกรอบความมั่นคงปลอดภัยไซเบอร์ NIST กับ วงจรชีวิตการจัดการช่องโหว่โดย CDC มาพัฒนากรอบการจัดการช่องโหว่ระบบสารสนเทศขององค์กร จึงทำให้เกิดกรอบการจัดการช่องโหว่ระบบสารสนเทศเพื่อการรักษาความมั่นคงปลอดภัยทางไซเบอร์ ของกองบัญชาการกองทัพภาคที่ 2

นนทวัฒน์ ทับทิม และ สุปกรณ์ สวัสดิ์พุทรา [15] วิเคราะห์และทำการรวบรวมช่องโหว่ที่พบในเครือข่ายมาแสดงผลสรุปเป็นรายงานโดยใช้เครื่องมือบนระบบปฏิบัติการ Kali Linux ซึ่งเครื่องมือที่ใช้ในการค้นหาและรวบรวมช่องโหว่ก็จะแตกต่างกันออกไปขึ้นอยู่กับประเภทของช่องโหว่ ขั้นตอนดำเนินงานได้แก่การค้นหาและรวบรวมช่องโหว่ที่พบในเครือข่ายโดยใช้เครื่องมือจากระบบปฏิบัติการ Kali Linux จากนั้นใช้ภาษาไพธอนในการค้นหาช่องโหว่อัตโนมัติและแสดงผลผ่านทางเว็บแอปพลิเคชัน และสุดท้ายได้ใช้เครื่องมือ Nmap ให้สามารถแจ้งเตือนผ่าน Line Official Account ได้ เพื่อให้สามารถใช้งานได้อย่างสะดวกสบายเพิ่มมากขึ้น

วิชสุนี ธีรรัชต์กาญจน์ [16] รวบรวม วิเคราะห์ และจำแนกรูปแบบการโจมตีไซเบอร์บนระบบปฏิบัติการLinux พฤติกรรมน่าสงสัยจะถูกรวบรวมผ่านทางฮันนีพอตที่ถูกติดตั้งไว้เป็นกับดักล่อ

เหยื่อผู้บุกรุกทางไซเบอร์ โดยข้อมูลจะถูกเก็บในรูปแบบของบันทึกจัดเก็บ หลังจากนั้นจะถูกสั่งการผ่านเชลล์สคริปต์เพื่อวิเคราะห์หารูปแบบการโจมตี จากการทดลองพบว่าการโจมตีที่พบมากที่สุดจะอยู่ในรูปแบบของ Malware

สุมาศ กันจินะ [17] เสนอวิธีการวิเคราะห์ และค้นหาช่องโหว่ รวมถึงวิธีการหลบเลี่ยงมาตรการรักษาความมั่นคงปลอดภัยของระบบเว็บแอปพลิเคชัน ซึ่งเป็นระบบที่ถูกพัฒนาด้วยภาษาไพธอน เพื่อช่วยวิเคราะห์หาช่องโหว่ของเว็บแอปพลิเคชัน โดยเฉพาะช่องโหว่ในรูปแบบของ Cross-site Scripting (XSS) attacks เพื่อทำให้ทีมพัฒนาระบบมีความรู้ความเข้าใจ และค้นพบข้อผิดพลาดตลอดจนสามารถวิเคราะห์มาตรการรักษาความมั่นคงปลอดภัย และการหลบหลีกการตรวจจับด้วยเทคนิคต่าง ๆ อย่างสม่ำเสมอ จนสามารถแก้ไขโดยวิธีที่ถูกต้อง ระบบสามารถวิเคราะห์และค้นหาช่องโหว่ Cross-site Scripting (XSS) attacks ในรูปแบบต่าง ๆ ที่สามารถหลบเลี่ยงมาตรการรักษาความมั่นคงปลอดภัยได้ สามารถทดสอบการโจมตีในรูปแบบของภัยคุกคามต่าง ๆ เพื่อสามารถเรียนรู้และหามาตรการป้องกัน เพื่อลดความเสี่ยงที่เกิดขึ้นกับการใช้งานเว็บแอปพลิเคชันขององค์กรนั้น ๆ ได้อย่างมีประสิทธิภาพ

งานวิจัยบางส่วนเน้นการนำเครื่องมือที่ศึกษามาใช้กำหนดกรอบและวิธีการตรวจสอบช่องโหว่ในองค์กร ตัวอย่างเช่น

ณัชณภัทร ใจอดทน [8] หาช่องโหว่หรือจุดอ่อนของเว็บไซต์กรมควบคุมการปฏิบัติทางอากาศว่ามีช่องโหว่หรือไม่ และวิเคราะห์ผลกระทบของช่องโหว่หรือจุดอ่อนนั้นว่ามีระดับความรุนแรงและมีผลกระทบต่อเว็บไซต์อย่างไร โดยใช้โปรแกรม Acunetic Web Vulnerability- Scanner เป็นเครื่องมือในการตรวจหาช่องโหว่ เพื่อหาแนวทางการแก้ไขและป้องกันการถูกโจมตีจากผู้ไม่หวังดีจากการใช้ประโยชน์จากช่องโหว่หรือจุดอ่อนของเว็บไซต์เพื่อใช้เป็นแนวทางในการพัฒนาให้มีความมั่นคงปลอดภัย ผลจากการสแกนพบว่ามีช่องโหว่ทั้งหมด 5 ช่องโหว่ โดยช่องโหว่ที่มีระดับความรุนแรงสูงประกอบด้วย File upload ซึ่งเป็นช่องโหว่ที่สามารถถูกใช้ประโยชน์โดยผู้ไม่หวังดีในการโจมตี

ปิยะ อาจหาญ [18] เจาะระบบเว็บแอปพลิเคชันด้วยเทคนิค SQL Injection ตลอดจนวิธีการหลบหลีกการตรวจจับและป้องกัน โดยเฉพาะอย่างยิ่ง SQL Injection ผ่าน HTTP Header ซึ่งเป็นเทคนิคหนึ่งที่ผู้บุกรุกสามารถใช้เป็นวิธีในการหลบหลีกการตรวจจับ โดยงานวิจัยนี้แสดงให้เห็นว่าการนำ Web Application Firewall (WAF) มาประยุกต์ใช้ร่วมกับ Stateful Firewall สามารถป้องกันการโจมตีดังกล่าวได้อย่างมีประสิทธิภาพ

เครื่องมือที่ใช้สแกนช่องโหว่ที่ถูกนำมาทดสอบและเปรียบเทียบเพื่อนำมาประยุกต์ใช้ในองค์กรมีมากมายหลายเครื่องมือ แต่ละเครื่องมือก็จะได้ผลลัพธ์แตกต่างกันไปตามข้อจำกัดหรือ

ขีดความสามารถของแต่ละเครื่องมือ เครื่องมือที่เลือกใช้จึงควรมีประสิทธิภาพและใช้งานได้ง่าย ซึ่งเครื่องมือที่เลือกใช้ก็จะช่วยเสริมสร้างความมั่นคงปลอดภัยองค์กรได้อย่างครอบคลุม โดยจากการวิเคราะห์ประสิทธิภาพของเครื่องมือที่นิยมใช้ในองค์กร พบว่า OpenVAS, Nessus และ OWASP ZAP เป็นเครื่องมือที่ได้รับความนิยมเนื่องจากเป็นเครื่องมือแบบ Open Source ที่สามารถใช้งานได้ฟรีโดยไม่มีค่าใช้จ่าย OpenVAS สามารถวิเคราะห์และประเมินระดับความรุนแรงของช่องโหว่ได้ มีผลลัพท์ที่ครอบคลุม ข้อมูลน่าเชื่อถือ ใช้งานง่าย มีคำแนะนำในการแก้ไขช่องโหว่ต่างๆเพื่อที่จะลดความเสี่ยงที่เกิดขึ้นลงได้ ส่วน Nessus มีความสามารถในการสแกนได้รวดเร็วเร็วมีความแม่นยำ เวลาในการสแกนน้อย และสามารถตรวจจับช่องโหว่ได้อย่างมีประสิทธิภาพ และ OWASP ZAP สามารถทำการทดสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน มีความยืดหยุ่นและมีคุณสมบัติที่หลากหลาย และสามารถทำงานได้บนหลายระบบปฏิบัติการ เช่น Linux Windows และอื่นๆ

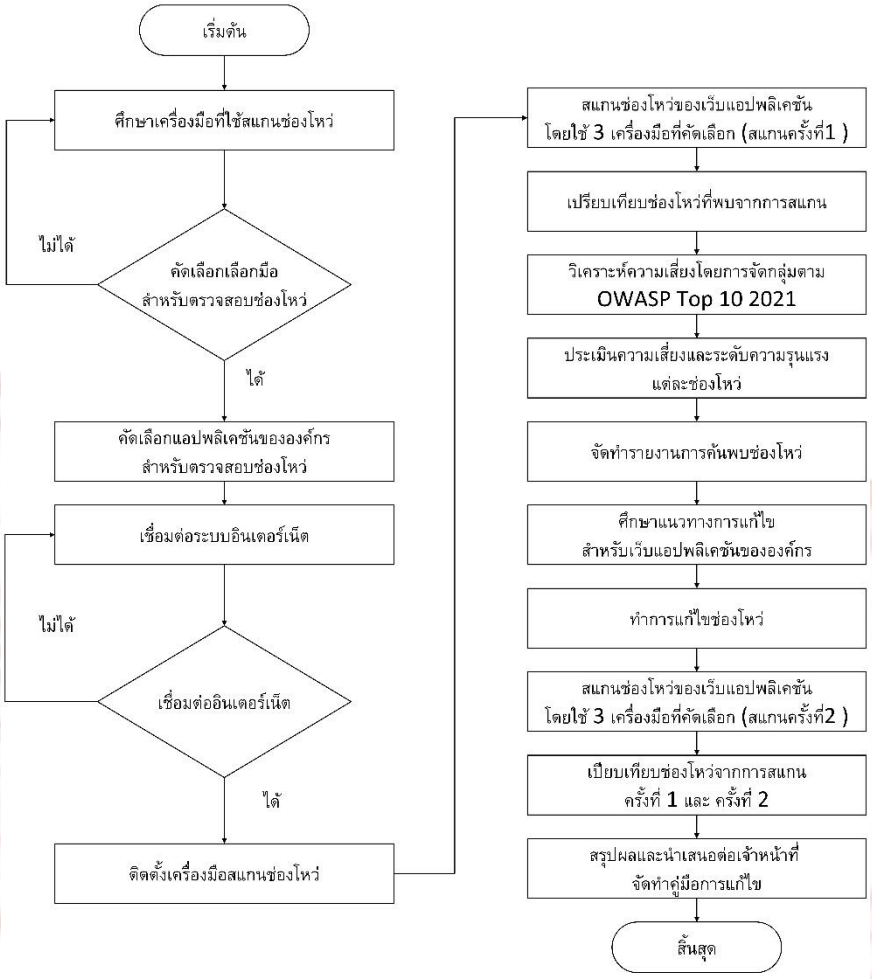
จากผลการศึกษาของงานวิจัยที่ได้กล่าวมาแล้ว ส่วนใหญ่เน้นการนำเครื่องมือเพียง 1-2 เครื่องมือมาใช้ในการตรวจสอบช่องโหว่หรือเปรียบเทียบผลลัพธ์การตรวจสอบช่องโหว่ของแต่ละเครื่องมือ หรือเลือกพัฒนาเครื่องมือสำหรับตรวจสอบช่องโหว่เอง ดังนั้น สำหรับการค้นคว้าอิสระนี้ที่มุ่งเน้นการศึกษาข้อจำกัดและประสิทธิภาพของเครื่องมือการตรวจสอบช่องโหว่เพื่อให้สามารถนำมาประยุกต์ใช้ในองค์กรซึ่งเป็นบริษัทเอกชนขนาดกลางแห่งหนึ่งได้ ผู้วิจัยจึงเลือกเครื่องมือที่นิยมใช้จำนวน 3 เครื่องมือ คือ OpenVAS, Nessus และ OWASP ZAP มาทดลองใช้ตรวจสอบช่องโหว่ในองค์กร เพื่อวิเคราะห์หาแนวทางการป้องกันช่องโหว่ที่เกิดจากการตั้งค่าระบบและช่องโหว่ที่อยู่บนเว็บแอปพลิเคชันขององค์กร

บทที่ 3
วิธีดำเนินการวิจัย

เว็บแอปพลิเคชันเป็นเครื่องมือสำคัญที่องค์กรใช้ในการดำเนินงานในปัจจุบัน อย่างไรก็ตาม การใช้งานเว็บแอปพลิเคชันที่เพิ่มขึ้นส่งผลให้มีความเสี่ยงด้านความมั่นคงปลอดภัยมากขึ้น ดังนั้น การตรวจสอบช่องโหว่ของเว็บแอปพลิเคชันจึงเป็นกระบวนการที่สำคัญและจำเป็นเพื่อป้องกันภัยคุกคามทางไซเบอร์และลดความเสี่ยงที่อาจเกิดขึ้น การค้นคว้าอิสระนี้มุ่งเน้นการเปรียบเทียบเครื่องมือที่ใช้ในการตรวจสอบช่องโหว่ โดยใช้บริษัทเอกชนขนาดกลางเป็นกรณีศึกษาในการทดสอบเครื่องมือ จึงมีกรอบแนวทางในการดำเนินการค้นคว้าอิสระ ดังนี้

3.1 ออกแบบขั้นตอนการวิจัย

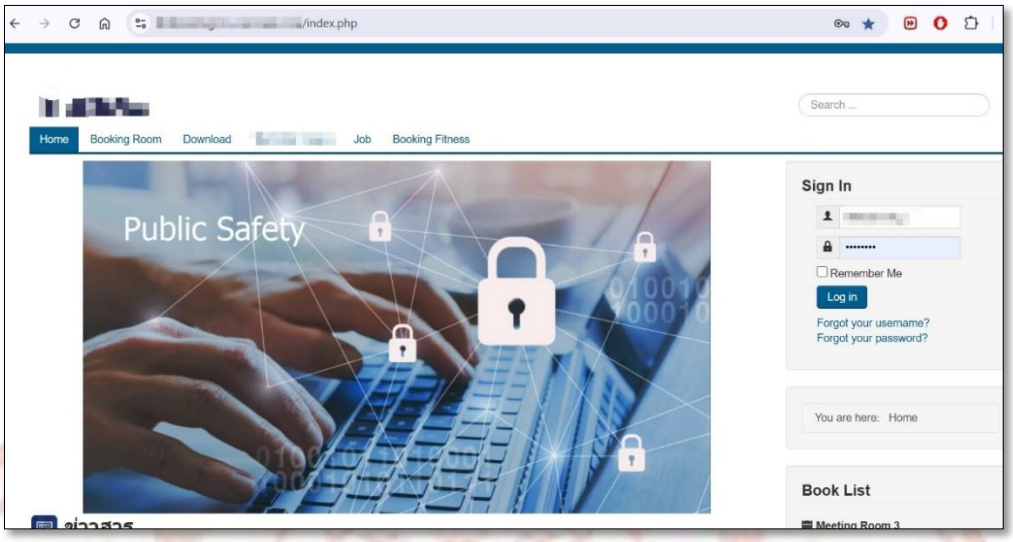
ผู้วิจัยได้มีการเริ่มต้นเข้าใช้งานเว็บแอปพลิเคชันขององค์กรอยู่เป็นประจำ และผู้วิจัยเห็นว่าเว็บแอปพลิเคชันดังกล่าวมีการใช้งานมาอย่างยาวนาน จึงควรมีการตรวจสอบช่องโหว่ของเว็บแอปพลิเคชันเพื่อความมั่นคงปลอดภัยที่เพิ่มมากขึ้น จึงได้ทำการศึกษาเครื่องมือต่างๆ ที่เกี่ยวข้องเพื่อใช้ในการสแกนหาช่องโหว่ โดยแบ่งการสแกนออกเป็น 2 รอบคือสแกนก่อนการแก้ไขช่องโหว่ที่ตรวจพบ และ สแกนหลังการแก้ไขช่องโหว่ที่ตรวจพบ เมื่อทำการสแกนรอบแรกเสร็จแล้ว แต่ละเครื่องจะต้องทำรายงานออกเพื่อนำข้อมูลที่ได้นำมาวิเคราะห์ โดยจัดกลุ่มรายการความเสี่ยงด้านความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน ตาม OWASP Top Ten 2021 หลังจากนั้นจัดทำรายงานเพื่อนำเสนอต่อผู้ที่มีส่วนเกี่ยวข้องหรือเจ้าหน้าที่ที่ดูแลระบบ เพื่อทำการแก้ไขช่องโหว่ที่มีระดับความรุนแรงที่มีความเสี่ยงสูงหรือช่องโหว่ที่สามารถทำได้ทันที จากนั้น ทำการสแกนซ้ำรอบที่ 2 เพื่อดูว่าช่องโหว่นั้นหายไปหรือไม่ หรือลดระดับความเสี่ยงลงหรือไม่ ซึ่งกระบวนการดังกล่าวนี้จะถูกนำมาสรุปเพื่อจัดทำเป็นคู่มือแนวทางการป้องกันแก้ไข ดังภาพที่ 3-1



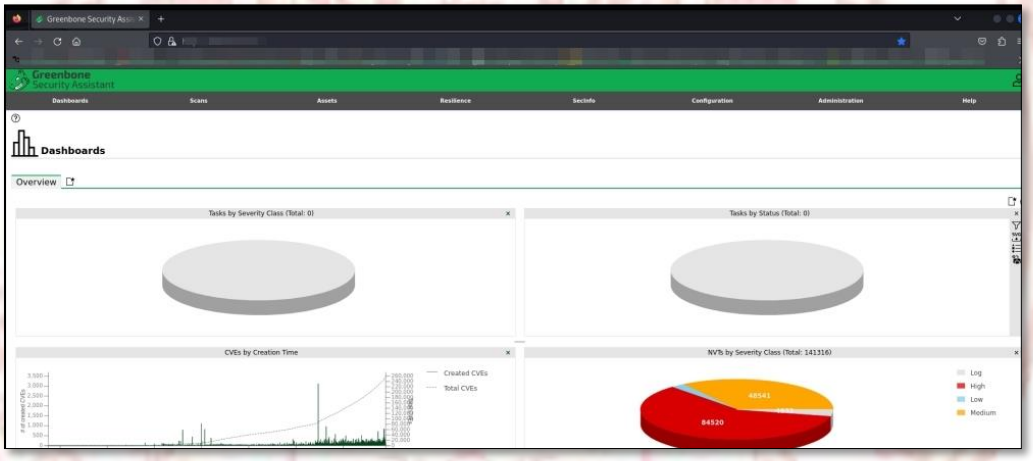
ภาพที่ 3-1 กรอบแนวทางการตรวจสอบช่องโหว่ในองค์กร

3.2 การสแกนช่องโหว่

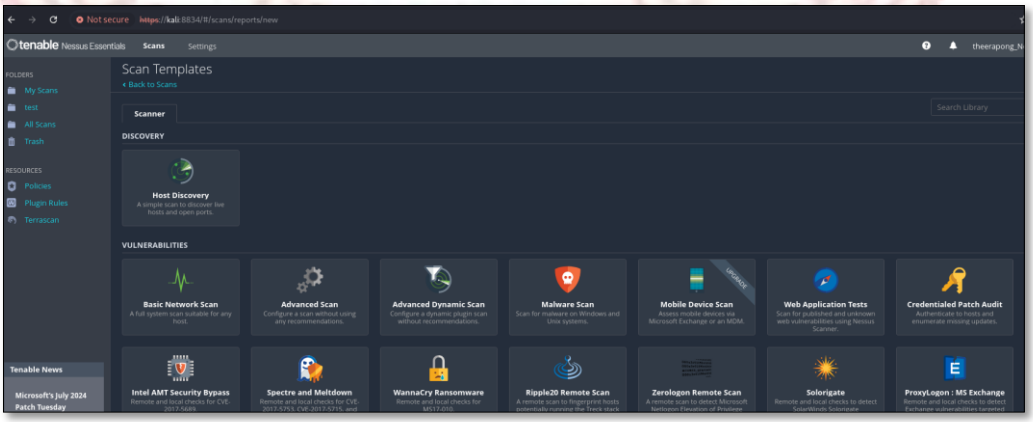
ผู้วิจัยได้ศึกษาเครื่องมือต่าง ๆ ที่เกี่ยวข้องเพื่อใช้ในการสแกนหาช่องโหว่ภายในองค์กรบนระบบปฏิบัติการ Linux และคัดเลือกเครื่องมือสแกนความมั่นคงปลอดภัยของเว็บแอปพลิเคชันที่ใช้สำหรับการประเมินช่องโหว่จำนวน 3 เครื่องมือ ได้แก่ OpenVAS, Nessus, OWASP ZAP โดยทดสอบกับเว็บแอบพลิชันขององค์ที่ที่มีการใช้งานจริง ซึ่งเป็นระบบประเภทเว็บไซต์การเข้าใช้งานทั่วไปภายในองค์กร เช่น การจองห้องประชุม การดาวน์โหลดเอกสาร และการจองฟิตเนส เป็นต้น ดังแสดงในภาพที่ 3-2 ถึง 3-5ผู้วิจัยได้ศึกษาขั้นตอนการโจมตีแบบฟิชชิง [37] เพื่อวางแผนในการจำลองสถานการณ์ในรูปแบบอีเมลฟิชชิง แก่บุคลากรภายในองค์กร เพื่อต้องการที่จะการประเมินการรับรู้ความปลอดภัยทางไซเบอร์ขององค์กรได้ โดยมีขั้นตอน ดังภาพที่ 3-2 และ 3-3



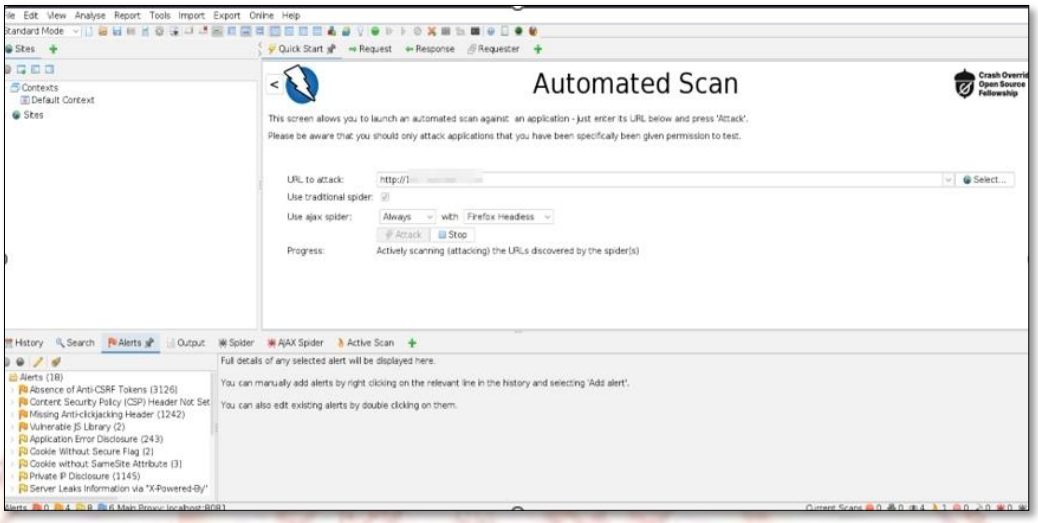
ภาพที่ 3-2 เว็บแอปพลิชันขององค์กร



ภาพที่ 3-3 หน้าหลักของโปรแกรม OpenVAS

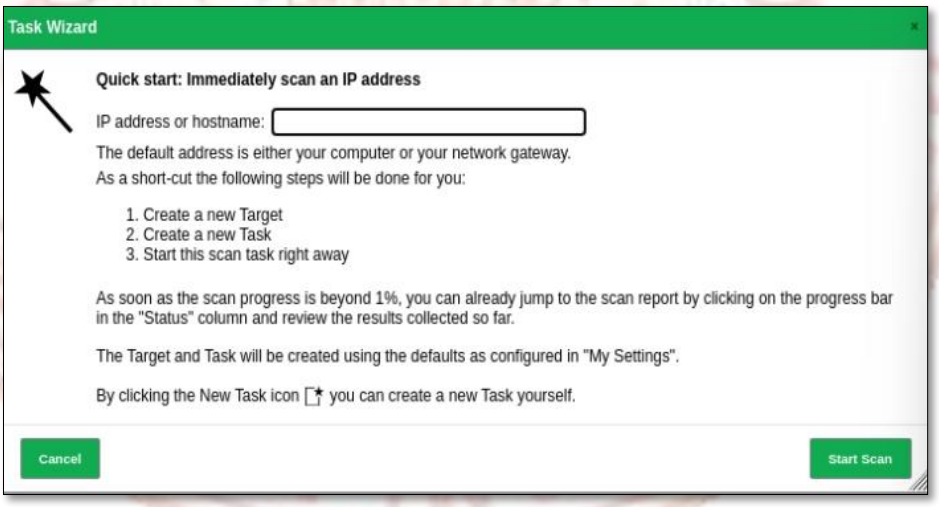


ภาพที่ 3-4 หน้าหลักของโปรแกรม Nessus

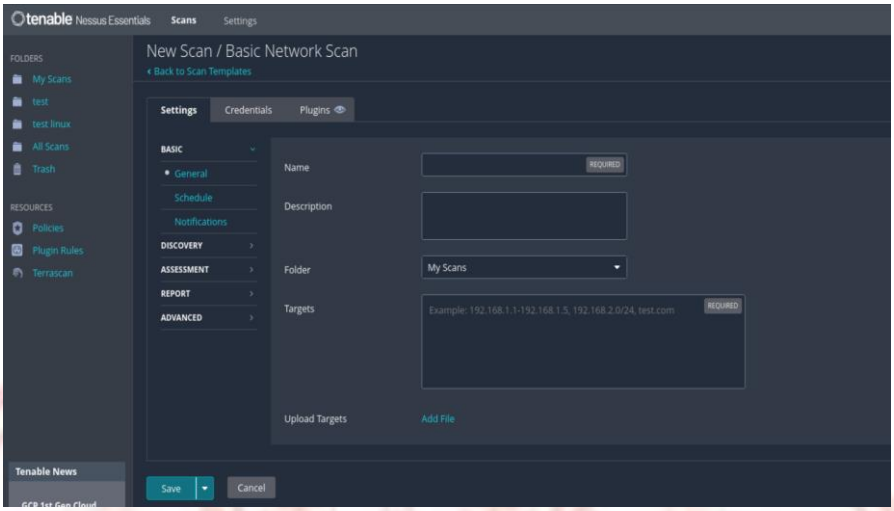


ภาพที่ 3-5 หน้าหลักของโปรแกรม OWASP ZAP

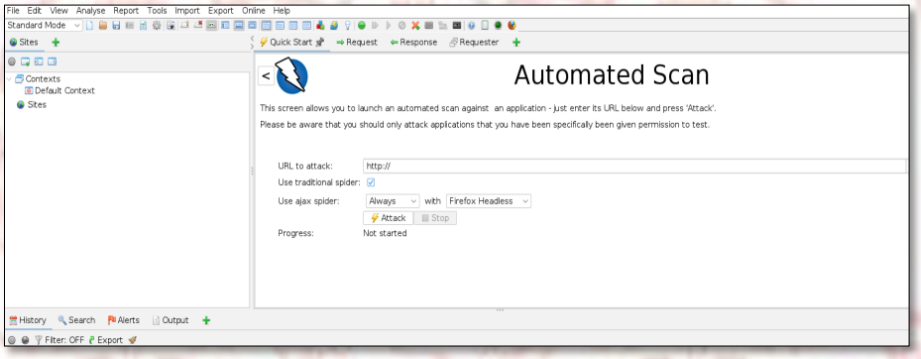
ผู้วิจัยได้ทำการสแกนช่องโหว่ของเว็บแอปพลิเคชันโดยการกรอก URLหรือ IP Address ของเป้าหมายในเครื่องมือที่คัดเลือกมาทั้งหมด 3 เครื่องมือ เพื่อตรวจสอบหาช่องโหว่ โดยการตั้งค่าเป้าหมายบนเครื่องมือทั้ง 3 เครื่องมือ แสดงดังภาพที่ 3-6 ถึง 3-8



ภาพที่ 3-6 การตั้งค่าการสแกนของโปรแกรม OpenVAS



ภาพที่ 3-7 การตั้งค่าการสแกนของโปรแกรม Nessus



ภาพที่ 3-8 การตั้งค่าการสแกนของโปรแกรม OWASP ZAP

3.3 การตรวจสอบและเปรียบเทียบช่องโหว่ที่ได้จากการสแกน

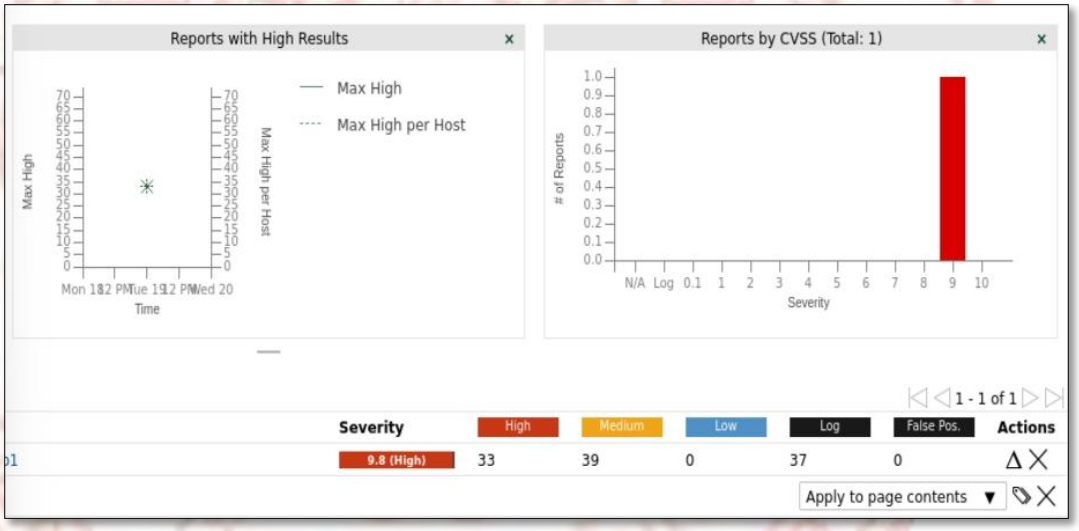
ผู้วิจัยได้ทำการคัดเลือกเครื่องมือสแกนความมั่นคงปลอดภัยของเว็บแอปพลิเคชันจำนวน 3 เครื่องมือ ได้แก่ OpenVAS, Nessus และ OWASP ZAP โดยทดสอบกับเว็บแอบพลิชันขององค์ที่ที่มีการใช้งานจริง ซึ่งเป็นระบบประเภทเว็บไซต์การเข้าใช้งานทั่วไปภายในองค์กรเช่น การจองห้องประชุม การดาวน์โหลดเอกสาร และการจองฟิตเนส เป็นต้น ผู้วิจัยจะดำเนินการสแกนช่องโหว่ของเว็บแอปพลิเคชันที่คัดเลือกไว้ จำนวน 2 รอบ โดยผลการสแกนที่ได้จากครั้งแรกจะถูกนำไปเปรียบเทียบช่องโหว่ทั้ง 3 เครื่องมือ เพื่อวิเคราะห์ความเสี่ยงจากช่องโหว่โดยการจัดกลุ่มตาม OWASP Top 10 2021 จัดทำรายงานการค้นพบช่องโหว่ให้ผู้ดูแลระบบเพื่อดำเนินการแก้ไข และเมื่อแก้ไขแล้วเสร็จ ได้ดำเนินการสแกนระบบรอบที่ 2 อีกรอบเพื่อตรวจสอบช่องโหว่ที่คงเหลือและจัดทำเป็นรายงานพร้อมแนวปฏิบัติขององค์กรในการบำรุงรักษาระบบให้มีความมั่นคงปลอดภัย

บทที่ 4
ผลการวิจัย

4.1 ผลการสแกนช่องโหว่ครั้งที่ 1

4.1.1 ผลการสแกนด้วย OpenVAS ครั้งที่ 1

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ OpenVAS ในการทดสอบการสแกนครั้งที่ 1 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 72 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีระดับความรุนแรงสูง 33 ช่องโหว่ และช่องโหว่ที่มีระดับความรุนแรงปานกลาง 39 ช่องโหว่ แสดงดังภาพที่ 4-1



ภาพที่ 4-1 ผลการสแกนของโปรแกรม OpenVAS (รอบที่ 1)

โดยระดับความรุนแรงของ CVSS v2.0 (Common Vulnerability Scoring System) ใช้เกณฑ์การให้คะแนนได้ 3 ระดับคือ

High (7.0 – 10.0) – มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย

Medium (4.0 - 6.9) – มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี

Low (0.0-3.9) – มีผลกระทบต่ำ มีความมั่นคงปลอดภัย ไม่สามารถโจมตีได้ง่าย

ตารางที่ 4-1 รายการช่องโหว่ที่ตรวจพบจากเครื่องมือ OpenVAS และระดับความรุนแรง

ช่องโหว่	ความรุนแรง
NVT: Joomla Multiple Vulnerabilities (20180507, 20180505, 20180504)	สูง (High)
NVT: Joomla! 3.0.0 - 3.9.22 SQL Injection Vulnerability	สูง (High)
NVT: Joomla! 3.0.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	สูง (High)
NVT: Joomla 3.7.0 <= 3.8.3 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! < 3.9.7 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 2.5.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 2.5.0 - 3.9.13 SQL Injection Vulnerability	สูง (High)
NVT: Joomla! 1.7.0 - 3.9.15 SQL Injection Vulnerability	สูง (High)
NVT: Joomla! < 3.9.3 Multiple Vulnerabilities	สูง (High)
NVT: Joomla < 3.8.12 Multiple Vulnerabilities	สูง (High)
NVT: Joomla < 3.9.5 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 3.2.0 - 3.9.24 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! Core 'PHP' Local File Inclusion Vulnerability (20180601)	สูง (High)
NVT: Joomla! < 3.8.13 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! < 3.9.13 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 3.0.0 - 3.9.14 Multiple Vulnerabilities	สูง (High)
NVT: Joomla 'User Notes list view' SQL Injection Vulnerability	สูง (High)
NVT: Joomla! 3.2.0 - 3.9.15 CSRF Vulnerability	สูง (High)
NVT: Joomla! 3.7.0 - 3.9.15 Access Control Vulnerability	สูง (High)
NVT: Joomla! 3.7.0 - 3.9.18 CSRF Vulnerability	สูง (High)
NVT: Joomla Multiple Vulnerabilities (20180502, 20180501)	สูง (High)
NVT: Joomla! < 3.8.13 ACL Violation Vulnerability	สูง (High)
NVT: Joomla! 3.0.0 - 3.9.24 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 2.5.0 - 3.9.27 Multiple Vulnerabilities	สูง (High)

ตารางที่ 4-1 (ต่อ)

ช่องโหว่	ความรุนแรง
NVT: Joomla! 2.5.0 - 3.9.18 Textlter Vulnerability	สูง (High)
NVT: Joomla < 3.8.12 ACL Violation Vulnerability	สูง (High)
NVT: Joomla! 2.5.0 - 3.9.15 Access Control Vulnerability	สูง (High)
NVT: Joomla! 1.6.0 - 4.4.0, 5.0.0 Information Disclosure Vulnerability	สูง (High)
NVT: Joomla! < 3.9.4 Multiple Vulnerabilities	สูง (High)
NVT: SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	สูง (High)
NVT: Joomla! 2.5.0 - 3.9.22 Multiple Vulnerabilities	สูง (High)
NVT: Joomla! 1.7.0 - 3.9.22 ACL Violation Vulnerability	สูง (High)
NVT: Joomla! < 3.8.13 RCE Vulnerability	สูง (High)
NVT: Joomla! 3.0.0 - 3.9.26 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla 'com_elds' RCE Vulnerability (20180506)	ปานกลาง (Medium)
NVT: Joomla! XSS Vulnerability (20240203)	ปานกลาง (Medium)
NVT: Joomla! Open Redirect Vulnerability (20240202)	ปานกลาง (Medium)
NVT: Joomla! Multiple Vulnerabilities (20240802, 20240805)	ปานกลาง (Medium)
NVT: Joomla! XSS Vulnerability (20240205)	ปานกลาง (Medium)
NVT: Joomla! URL Validation Vulnerability (20240801)	ปานกลาง (Medium)
NVT: Joomla! <= 3.9.19 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! < 3.9.12 XSS Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 1.7.0 <= 3.9.5 XSS Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 2.5.0 - 3.9.24 Multiple XSS Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.15 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.18 Multiple XSS Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.20 Open Redirect Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.25 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.27 Multiple XSS Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! 3.1.0 - 3.9.23 XSS Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 3.7.0 - 3.10.6 XSS Vulnerability	ปานกลาง (Medium)

ตารางที่ 4-1 (ต่อ)

ช่องโหว่	ความรุนแรง
NVT: Joomla! < 3.9.2 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla 'Chromes' module XSS Vulnerability	ปานกลาง (Medium)
NVT: Joomla 'Language Switcher' Module Cross Site Scripting Vulnerability (20180602)	ปานกลาง (Medium)
NVT: Joomla 'Media Manager' XSS Vulnerability (20180509)	ปานกลาง (Medium)
NVT: Joomla! Multiple XSS Vulnerabilities (20240703, 20240704)	ปานกลาง (Medium)
NVT: Joomla 'Uri' class XSS Vulnerability	ปานกลาง (Medium)
NVT: Joomla! XSS Vulnerability (20240705)	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.20 Directory Traversal Vulnerability	ปานกลาง (Medium)
NVT: HTTP Debugging Methods (TRACE/TRACK) Enabled	ปานกลาง (Medium)
NVT: Joomla! Session Expiration Vulnerability (20240201)	ปานกลาง (Medium)
NVT: Joomla! 3.8.0 - 3.9.13 Path Disclosure Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 3.0.0 - 3.9.23 ACL Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 2.5.0 - 3.9.16 Multiple Vulnerabilities	ปานกลาง (Medium)
NVT: Joomla! < 3.9.11 Mail Submission Vulnerability	ปานกลาง (Medium)
NVT: Joomla! 1.6.0 - 3.9.24 ACL Vulnerability	ปานกลาง (Medium)
NVT: Missing 'Secure' Cookie Attribute (HTTP)	ปานกลาง (Medium)
NVT: Joomla 'Redirect' Method XSS Vulnerability (20180508)	ปานกลาง (Medium)
NVT: Joomla 'Unpublished Tags' Information Disclosure Vulnerability	ปานกลาง (Medium)
NVT: Joomla! < 3.8.13 Access Level Violation Vulnerability	ปานกลาง (Medium)
NVT: SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	ปานกลาง (Medium)
NVT: HTTP Debugging Methods (TRACE/TRACK) Enabled	ปานกลาง (Medium)

จากผลการตรวจสอบช่องโหว่โดย OpenVAS ดังแสดงในตารางที่ 4-1 พบว่ามีรายการช่องโหว่หลายรายการที่คล้ายกัน เช่น ช่องโหว่ประเภท Cross-Site Scripting (XSS) มีปรากฏอยู่ถึง 15 รายการ ซึ่งจากการวิเคราะห์รายละเอียดของแต่ละช่องโหว่ พบว่าแต่ละรายการเป็นช่องโหว่ XSS ที่ปรากฏอยู่หลายแห่งในระบบ รายละเอียดดังแสดงในตารางที่ 4-2

ตารางที่ 4-2 รายละเอียดของช่องโหว่ Cross-Site Scripting (XSS) ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla 'Chromes' module XSS Vulnerability	การไม่ทำการ escape ใน โมดูล chromes ของ Joomla! ทำให้เกิดช่องโหว่ Cross-Site Scripting (XSS)
2	NVT: Joomla 'Media Manager' XSS Vulnerability (20180509)	การกรองชื่อไฟล์และโฟลเดอร์ใน Media Manager ที่ไม่มีประสิทธิภาพ
3	NVT: Joomla 'Redirect' Method XSS Vulnerability (20180508)	การไม่ทำการ escape ข้อมูลในส่วนของ user-info component ของ URI
4	NVT: Joomla 'Uri' class XSS Vulnerability	การกรองข้อมูลอินพุตที่ไม่เพียงพอในคลาส Uri (เดิมชื่อ JUri)
5	NVT: Joomla! < 3.9.12 XSS Vulnerability	การไม่ทำการ escape ซึ่งเปิดโอกาสให้ผู้โจมตีสามารถใช้ logo parameter ของ default templates เพื่อแทรกโค้ดอันตราย
6	NVT: Joomla! 1.7.0 <= 3.9.5 XSS Vulnerability	การไม่ทำการ escape ข้อมูลที่ผู้ใช้ป้อนอย่างถูกต้อง ใน debug views ของ com_users ทำให้ผู้โจมตีสามารถฝังโค้ด JavaScript อันตรายลงในหน้าเว็บได้
7	NVT: Joomla! 2.5.0 - 3.9.24 Multiple XSS Vulnerabilities	XSS ในข้อความแจ้งเตือน (Alert Messages) ช่องโหว่นี้เปิดโอกาสให้ผู้โจมตีแทรกโค้ดJavaScript ที่เป็นอันตรายผ่านข้อความแจ้งเตือน

ตารางที่ 4-2 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
8	NVT: Joomla! 3.0.0 - 3.9.18 Multiple XSS Vulnerabilities	XSS ใน modules heading tag option CVE-2020-13761: ช่องโหว่นี้เกิดจากการจัดการข้อมูลในตัวเลือก heading tag ของโมดูลไม่ปลอดภัยผู้โจมตีสามารถแทรกโค้ดอันตรายผ่านตัวเลือก heading tag XSS ใน com_modules tag options CVE-2020-13762: การจัดการข้อมูล tag options ของ com_modules ไม่เหมาะสม ผู้โจมตีสามารถแทรกโค้ดอันตรายใน tag options
9	NVT: Joomla! 3.0.0 - 3.9.27 Multiple XSS Vulnerabilities	CVE-2021-26035: การไม่ทำการ escape ข้อมูลใน rules field ของ J Form API ทำให้เกิดช่องโหว่ XSS ผู้โจมตีสามารถแทรกโค้ดอันตรายผ่าน field ที่ได้รับผลกระทบ CVE-2021-26039: การไม่ทำการ escape ข้อมูลใน image list view ของ com_media ทำให้เกิดช่องโหว่ XSS ส่งผลให้ผู้โจมตีสามารถแทรกโค้ด JavaScript ที่เป็นอันตรายในส่วนจัดการสื่อ
10	NVT: Joomla! 3.1.0 - 3.9.23 XSS Vulnerability	การไม่ทำการ escape พารามิเตอร์ที่เกี่ยวข้องกับรูปภาพในหลาย ๆ มุมมองของ com_tags ส่งผลให้เกิดช่องโหว่ XSS ซึ่งช่วยให้ผู้โจมตีสามารถแทรกโค้ด JavaScript ที่เป็นอันตรายลงในเนื้อหา
11	NVT: Joomla! 3.7.0 - 3.10.6 XSS Vulnerability	การขาดการตรวจสอบความถูกต้องของข้อมูลอินพุต (Input Validation) ใน com_fields
12	NVT: Joomla! XSS Vulnerability (20240203)	การตรวจสอบข้อมูลอินพุตใน media selection fields ไม่เพียงพอ ช่องโหว่นี้ส่งผลให้เกิดการโจมตีแบบ XSS ในหลายส่วนของส่วนขยาย

ตารางที่ 4-2 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
13	NVT: Joomla! Multiple XSS Vulnerabilities (20240703, 20240704)	CVE-2024-21731: ช่องโหว่ XSS ใน String Helper::truncate method CVE-2024-26278: ช่องโหว่ XSS ใน Wrapper extensions ช่องโหว่เหล่านี้เกิดจากการกรองข้อมูลไม่เพียงพอใน String Helper::truncate method และ Wrapper extensions ทำให้ผู้โจมตีสามารถแทรก JavaScript ที่เป็นอันตรายและรันในบริบทของผู้ใช้
14	NVT: Joomla! XSS Vulnerability (20240205)	การกรองเนื้อหาไม่เพียงพอในส่วนประกอบต่าง ๆ (Components)
15	NVT: Joomla! XSS Vulnerability (20240705)	มีช่องโหว่ Cross-Site Scripting (XSS) ใน media selection fields ซึ่งเป็นช่องทางที่ผู้โจมตีสามารถแทรกโค้ดอันตรายเพื่อรันบนเว็บเบราว์เซอร์ของผู้ใช้

นอกจากช่องโหว่ประเภท Cross-Site Scripting (XSS) OpenVAS ยังพบช่องโหว่ประเภทอื่นที่คล้ายกันอีกหลายรายการ ดังนี้ ช่องโหว่ประเภท ACL Violation มีปรากฏอยู่ 5 รายการ ดังแสดงในตารางที่ 4-3 ช่องโหว่ประเภท SQL Injection มีปรากฏอยู่ 4 รายการ ดังแสดงในตารางที่ 4-4 ช่องโหว่ประเภท Cross-Site Request Forgery (CSRF) มีปรากฏอยู่ 2 รายการ ดังแสดงในตารางที่ 4-5 ช่องโหว่ประเภท Open Redirect มีปรากฏอยู่ 2 รายการ ดังแสดงในตารางที่ 4-6 ช่องโหว่ประเภท RCE มีปรากฏอยู่ 2 รายการ ดังแสดงในตารางที่ 4-7 ช่องโหว่ประเภท Access Control มีปรากฏอยู่ 2 รายการ ดังแสดงในตารางที่ 4-8 และช่องโหว่ประเภท Multiple Vulnerabilities มีปรากฏอยู่ 24 รายการ ดังแสดงในตารางที่ 4-9

ตารางที่ 4-3 รายละเอียดของช่องโหว่ ACL Violation ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla < 3.8.12 ACL Violation Vulnerability	ฟีเจอร์ Custom Fields มีการตรวจสอบที่ไม่เพียงพอเกี่ยวกับฟิลด์ที่ถูกปิดใช้งาน
2	NVT: Joomla! < 3.8.13 ACL Violation Vulnerability	หากผู้โจมตีสามารถเข้าถึงบัญชีอีเมลของผู้ใช้ที่มีสิทธิ์อนุมัติการยืนยันตัวตนของผู้ดูแลระบบในกระบวนการลงทะเบียนได้ ผู้โจมตีจะสามารถเปิดใช้งานบัญชีของตนเองในฐานะผู้ดูแลระบบได้
3	NVT: Joomla! 1.6.0 - 3.9.24 ACL Vulnerability	การกรองเนื้อหาในฟอร์มไม่เพียงพอ อาจเปิดโอกาสให้มีการเขียนทับข้อมูลในช่อง "ผู้เขียน" (author field) ได้ โดยคอมโพเนนต์หลักที่ได้รับผลกระทบได้แก่: com_fields, com_categories, com_banners, com_contact, com_newsfeeds และ com_tags
4	NVT: Joomla! 1.7.0 - 3.9.22 ACL Violation Vulnerability	ขาดการตรวจสอบความถูกต้องของข้อมูลขาเข้า ขณะจัดการชุดกฎ ACL อาจนำไปสู่การละเมิดสิทธิ์ในการเขียน
5	NVT: Joomla! 3.0.0 - 3.9.23 ACL Vulnerability	ขาดการตรวจสอบสิทธิ์การเข้าถึง (ACL) ใน endpoint orderPosition ของ com_modules อาจทำให้มีการเปิดเผยชื่อของโมดูลที่ยังไม่ได้เผยแพร่หรือผู้ใช้ไม่มีสิทธิ์เข้าถึงได้

ตารางที่ 4-4 รายละเอียดของช่องโหว่ SQL Injection ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla 'User Notes list view' SQL Injection Vulnerability	ช่องโหว่นี้เกิดจากการที่ Joomla ไม่ได้ทำการแปลงประเภทของตัวแปร อย่างเหมาะสมในคำสั่ง SQL ภายในมุมมองรายการบันทึกของผู้ใช้
2	NVT: Joomla! 1.7.0 - 3.9.15 SQL Injection Vulnerability	การไม่เปลี่ยนประเภทของตัวแปร ในคำสั่ง SQL ส่งผลให้เกิดช่องโหว่ SQL Injection ในเมนูประเภท "Featured Articles" บนส่วนแสดงผลด้านหน้าเว็บไซต์
3	NVT: Joomla! 2.5.0 - 3.9.13 SQL Injection Vulnerability	การขาดการตรวจสอบค่าพารามิเตอร์การกำหนดค่าที่ใช้ในคำสั่ง SQL ส่งผลให้เกิดช่องโหว่ SQL Injection ได้หลายรูปแบบ
4	NVT: Joomla! 3.0.0 - 3.9.22 SQL Injection Vulnerability	การกำหนดค่ารายการกรองแบบ Blacklist อย่างไม่เหมาะสม ส่งผลให้เกิดช่องโหว่ SQL Injection ในหน้ารายการผู้ใช้ ของระบบหลังบ้าน

ตารางที่ 4-5 รายละเอียดของช่องโหว่ Cross-Site Request Forgery ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla! 3.2.0 - 3.9.15 CSRF Vulnerability	ขาดการตรวจสอบโทเค็นในกระบวนการจัดการภาพของ com_templates ส่งผลให้เกิดช่องโหว่ Cross-Site Request Forgery (CSRF)
2	NVT: Joomla! 3.7.0 - 3.9.18 CSRF Vulnerability	Joomla! มีความเสี่ยงจากช่องโหว่ Cross-Site Request Forgery (CSRF) เนื่องจากไม่มีการตรวจสอบโทเค็น (Token) ใน com_postinstall

ตารางที่ 4-6 รายละเอียดของช่องโหว่ Open Redirect ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla! 3.0.0 - 3.9.20 Open Redirect Vulnerability	การขาดการตรวจสอบความถูกต้องของข้อมูลที่ป้อนเข้ามาใน com_content ส่งผลให้เกิดช่องโหว่การเปลี่ยนเส้นทางโดยไม่ปลอดภัย (Open Redirect)
2	NVT: Joomla! Open Redirect Vulnerability (20240202)	การแยกวิเคราะห์ URL อย่างไม่เหมาะสม (Inadequate parsing) อาจนำไปสู่ช่องโหว่การเปลี่ยนแปลงเส้นทางโดยไม่ปลอดภัย (Open Redirect)

ตารางที่ 4-7 รายละเอียดของช่องโหว่ RCE ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla! < 3.8.13 RCE Vulnerability	com_joomlaupdate มีความเสี่ยงที่เปิดโอกาสให้มีการเรียกใช้โค้ดโดยไม่ได้รับอนุญาต (Arbitrary Code-Execution) โดยเกิดจากการกำหนดค่าเริ่มต้นของ ACL ที่อนุญาตให้ผู้ใช้ระดับ Administrator เข้าถึงและดำเนินการกับ com_joomlaupdate ได้โดยตรง
2	NVT: Joomla! 'com_fields' RCE Vulnerability (20180506)	ช่องโหว่นี้เกิดจากการกรองข้อมูลในฟิลด์ filter ของ com_fields ที่ไม่เพียงพอหรือไม่เหมาะสม

ตารางที่ 4-8 รายละเอียดของช่องโหว่ Access Control ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla! 2.5.0 - 3.9.15 Access Control Vulnerability	การดำเนินการต่าง ๆ ภายใน com_templates ขาดการตรวจสอบสิทธิ์การเข้าถึง (ACL) ที่จำเป็น ซึ่งอาจเปิดช่องให้เกิดเวกเตอร์การโจมตีในรูปแบบต่าง ๆ ได้
2	NVT: Joomla! 3.7.0 - 3.9.15 Access Control Vulnerability	การควบคุมสิทธิ์การเข้าถึง (Access Control) ในฟิลด์ประเภท SQL ของ com_fields ไม่ถูกต้อง ส่งผลให้ผู้ใช้ที่ไม่ใช่ superadmin สามารถเข้าถึงได้โดยไม่ได้รับอนุญาต

ตารางที่ 4-9 รายละเอียดของช่องโหว่ Multiple Vulnerabilities ที่พบโดย OpenVAS

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
1	NVT: Joomla < 3.8.12 Multiple Vulnerabilities	การกรองข้อมูลขาออก บนหน้าข้อมูลโปรไฟล์ผู้ใช้ไม่เพียงพอ อาจนำไปสู่การโจมตีแบบ Stored XSS (CVE-2018-15880) และการตรวจสอบที่ไม่เหมาะสมในคลาส Input Filter อาจเปิดทางให้ไฟล์ PHAR ที่เตรียมไว้อย่างเฉพาะเจาะจงสามารถผ่านตัวกรองการอัปโหลดได้(CVE-2018-15882)
2	NVT: Joomla < 3.9.5 Multiple Vulnerabilities	คอมโพเนนต์ Media Manager ไม่ได้กรอกค่าพารามิเตอร์ folder อย่างถูกต้อง ทำให้ผู้โจมตีสามารถดำเนินการนอกเหนือจากไดเรกทอรีหลักของ Media Manager ได้ และจุดเชื่อมต่อ “รีเฟรชรายการเว็บไซต์ช่วยเหลือ” ของ com_users ขาดการตรวจสอบสิทธิ์การเข้าถึง ทำให้สามารถถูกเรียกใช้งานได้โดยผู้ใช้ที่ยังไม่ได้รับการยืนยันตัวตน
3	NVT: Joomla 3.7.0 <= 3.8.3 Multiple Vulnerabilities	การไม่แปลงประเภทของตัวแปรอย่างเหมาะสมในคำสั่ง SQL ภายในข้อความ post install ของเทมเพลต Hathor และการกรองข้อมูลเข้าที่ไม่เพียงพอใน com_fields สำหรับฟิลด์ประเภทต่าง ๆ เช่น list, radio และ checkbox
4	NVT: Joomla Multiple Vulnerabilities (20180502, 20180501)	ความผิดพลาดที่ทำให้ไฟล์ PHAR อาจถูกประมวลผลเป็นสคริปต์ PHP โดยเว็บเซิร์ฟเวอร์ ขึ้นอยู่กับการตั้งค่าของเซิร์ฟเวอร์ และการตรวจสอบสิทธิ์ในการเข้าถึง (Access Level Permissions) ที่ไม่เพียงพอ
5	NVT: Joomla Multiple Vulnerabilities (20180507, 20180505, 20180504)	Race Condition ที่ทำให้เซสชันซึ่งควรถูกทำลายไปแล้วกลับสามารถถูกสร้างขึ้นใหม่ได้ การกรองข้อมูลขาเข้า ที่ไม่เพียงพอ และความผิดพลาดในขั้นตอนติดตั้งผ่านเว็บ (Web Install Application)

ตารางที่ 4-9 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
6	NVT: Joomla! < 3.8.13 Multiple Vulnerabilities	การป้องกัน CSRF ที่ไม่เพียงพอใน com_installer (CVE-2018-17858) และ การตรวจสอบที่ไม่เพียงพอใน com_contact (CVE-2018-17859) อนุญาตให้มีการส่งอีเมลผ่านฟอร์มที่ถูกปิดใช้งาน
7	NVT: Joomla! < 3.9.13 Multiple Vulnerabilities	การขาดการตรวจสอบความถูกต้องใน com_template ส่งผลให้เกิดช่องโหว่ Cross-Site Request Forgery (CSRF) และการขาดการตรวจสอบสิทธิ์การเข้าถึงในไฟล์ mapping ของ phputf8 อาจนำไปสู่การเปิดเผยโครงสร้างเส้นทางของไฟล์ (Path Disclosure)
8	NVT: Joomla! 2.5.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	CVE-2022-23795: แถวข้อมูลของผู้ใช้ (User row) ไม่ได้เชื่อมโยงกับกลไกการยืนยันตัวตนอย่างเหมาะสม และ CVE-2022-23798: การตรวจสอบความถูกต้องของ URL ภายใน (Internal URLs) ไม่เพียงพอ
9	NVT: Joomla! 3.0.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	CVE-2022-23793: ช่องโหว่ Zip Slip ภายในตัวแยกไฟล์ .tar CVE-2022-23794: การเปิดเผยเส้นทางของระบบไฟล์ ผ่านข้อความแสดงข้อผิดพลาด และCVE-2022-23797: การกรองข้อมูลของ ID ที่เลือกไว้ไม่เพียงพอ
10	NVT: Joomla! < 3.9.4 Multiple Vulnerabilities	เค้าโครง item_title ในหน้าแก้ไข (edit views) ไม่มีการ escape ข้อมูลอย่างเหมาะสม ส่งผลให้เกิดช่องโหว่ XSS (CVE-2019-9711) ตัวจัดการ JSON ใน com_config ขาดการตรวจสอบข้อมูลขาเข้า ส่งผลให้เกิดช่องโหว่ XSS (CVE-2019-9712) ปลั๊กอินตัวอย่างข้อมูล ไม่มีการตรวจสอบสิทธิ์การเข้าถึง ส่งผลให้ผู้ใช้ที่ไม่ได้รับอนุญาตสามารถเข้าถึงได้ (CVE-2019-9713) และฟิลด์ฟอร์มของ Media ขาดการ escape ข้อมูล ส่งผลให้เกิดช่องโหว่ XSS (CVE-2019-9714)

ตารางที่ 4-9 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
11	NVT: Joomla! < 3.9.3 Multiple Vulnerabilities	การตั้งค่า "No Filtering" ใน Text Filter สามารถเขียนทับการตั้งค่าของลูกใน Global Configuration ได้ (CVE-2019-7739) การจัดการพารามิเตอร์ใน JavaScript (core.js, ฟังก์ชัน writeDynaList) ไม่เหมาะสม อาจทำให้เกิดช่องโหว่ XSS ได้ (CVE-2019-7740) การตรวจสอบที่ไม่เพียงพอใน Help URL Settings ของ Global Configuration เปิดโอกาสให้เกิด Stored XSS ได้ (CVE-2019-7741) การโจมตีแบบ XSS สามารถเกิดขึ้นได้จากการตั้งค่าเฉพาะของ Web Server ร่วมกับประเภทไฟล์บางชนิดและการวิเคราะห์ MIME Type ฝั่งเบราว์เซอร์ (CVE-2019-7742) ตัวจัดการ phar:// stream wrapper อาจถูกใช้โจมตีแบบ Object Injection ได้ หากไม่มีการป้องกัน เช่น การบล็อกหรือควบคุม phar:// handler สำหรับไฟล์ .phar (CVE-2019-7743) และการกรองข้อมูลในช่อง URL ของหลายคอมโพเนนต์หลักไม่เพียงพอ ซึ่งอาจนำไปสู่ช่องโหว่ XSS ได้ (CVE-2019-7744)
12	NVT: Joomla! < 3.9.7 Multiple Vulnerabilities	URL ของเซิร์ฟเวอร์อัปเดตใน com_joomlaupdate สามารถถูกแก้ไขโดยผู้ใช้ที่ไม่ใช่ Super Admin ได้ ประเภทฟิลด์ subform ไม่มีการกรองหรือการตรวจสอบความถูกต้องของข้อมูลใน subfields อย่างเพียงพอ ซึ่งอาจนำไปสู่ช่องโหว่ XSS และการส่งออก CSV จาก com_actionlogs มีช่องโหว่ที่สามารถถูกใช้โจมตีแบบ CSV Injection ได้

ตารางที่ 4-9 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
13	NVT: Joomla! 2.5.0 - 3.9.22 Multiple Vulnerabilities	CVE-2020-35610: คอมโพเนนต์ com_finder ไม่ตรวจสอบระดับการเข้าถึง ในฟีเจอร์แนะนำอัตโนมัติ CVE-2020-35611: การเปิดเผยข้อมูลสำคัญในหน้า Global Configuration CVE-2020-35612: ช่องโหว่ Path Traversal ในโมดูล mod_random_image และ CVE-2020-35615: ช่องโหว่ CSRF ในฟีเจอร์ส่งออกอีเมลของ com_privacy
14	NVT: Joomla! 2.5.0 - 3.9.27 Multiple Vulnerabilities	CVE-2021-26036: การขาดการตรวจสอบความถูกต้องของข้อมูลนำเข้า (Input Validation) อาจส่งผลให้ตารางกลุ่มผู้ใช้ เสียหาย CVE-2021-26037: ฟังก์ชันใน CMS ไม่ได้ยุติเซสชันของผู้ใช้ที่มีอยู่เดิมอย่างถูกต้อง เมื่อมีการเปลี่ยนรหัสผ่านหรือบัญชีผู้ใช้ถูกระงับ และCVE-2021-26038: การดำเนินการติดตั้ง (Install action) ใน com_installer ขาดการตรวจสอบสิทธิ์ (ACL) สำหรับ super users อย่างเหมาะสม
15	NVT: Joomla! 3.0.0 - 3.9.14 Multiple Vulnerabilities	CVE-2020-8419: การขาดการตรวจสอบโทเค็น ในการดำเนินการแบบกลุ่ม (Batch Actions) ของคอมโพเนนต์ต่าง ๆ และCVE-2020-8420: การขาดการตรวจสอบ CSRF Token ในตัวคอมไพเลอร์ LESS ของ com_templates ส่งผลให้เกิดช่องโหว่ CSRF
16	NVT: Joomla! 3.0.0 - 3.9.24 Multiple Vulnerabilities	CVE-2021-26027: การละเมิดสิทธิ์การเข้าถึง (ACL-Violation) ภายใน com_content สำหรับการแก้ไขเนื้อหาจากส่วนหน้าของเว็บไซต์ (Frontend Editing) CVE-2021-26028: ช่องโหว่ Path Traversal ในคลาส joomla/archive zip ซึ่งอาจเปิดทางให้เข้าถึงไฟล์หรือโฟลเดอร์นอกเส้นทางที่กำหนด และCVE-2021-23132: com_media อนุญาตให้เข้าถึงเส้นทาง (Paths) ที่ไม่ได้ตั้งใจให้ใช้สำหรับไฟล์รูปภาพ

ตารางที่ 4-9 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
17	NVT: Joomla! 3.2.0 - 3.9.24 Multiple Vulnerabilities	CVE-2021-23126, CVE-2021-23127: การสร้างรหัสลับของระบบยืนยันตัวตนแบบสองขั้นตอน (2FA) ใช้กลไกสุ่มที่ไม่ปลอดภัย CVE-2021-23128: ความเป็นไปได้ของการใช้กลไกสุ่มที่ไม่ปลอดภัยใน FOFEncryptRandval และCVE-2021-23131: การตรวจสอบข้อมูลนำเข้าภายในตัวจัดการเทมเพลต (Template Manager) ไม่เพียงพอ
18	NVT: Joomla! < 3.9.2 Multiple Vulnerabilities	CVE-2019-6261: การ escape ข้อมูลไม่เพียงพอใน com_contact CVE-2019-6262: การตรวจสอบค่าการตั้งค่า Help URL CVE-2019-6263: การตรวจสอบค่าการตั้งค่า Text Filter ใน Global Configuration ไม่เพียงพอ และCVE-2019-6264: การ escape ข้อมูลไม่เพียงพอใน mod_banners ทั้งหมดทำให้เกิดช่องโหว่ Stored XSS
19	NVT: Joomla! 2.5.0 - 3.9.16 Multiple Vulnerabilities	CVE-2020-11889: การตรวจสอบสิทธิ์การเข้าถึง (ACL) ในส่วนของระดับการเข้าถึงของ com_users ไม่ถูกต้อง ทำให้สามารถลบกลุ่มผู้ใช้ โดยไม่ได้รับอนุญาต และ CVE-2020-11890: การตรวจสอบข้อมูลนำเข้าในคลาสตารางกลุ่มผู้ใช้ ไม่เหมาะสม อาจส่งผลให้การกำหนดค่า ACL เสียหาย
20	NVT: Joomla! 3.0.0 - 3.9.15 Multiple Vulnerabilities	CVE-2020-10240: การขาดการตรวจสอบความยาวของข้อมูลในตารางผู้ใช้ อาจนำไปสู่การสร้างบัญชีผู้ใช้ที่มีชื่อผู้ใช้หรืออีเมลซ้ำกันได้ และCVE-2020-10242: การจัดการตัวเลือก CSS ที่ไม่เหมาะสมในธีม Protostar และ Beez3 ร่วมกับ JavaScript อาจเปิดช่องให้เกิดการโจมตีแบบ XSS

ตารางที่ 4-9 (ต่อ)

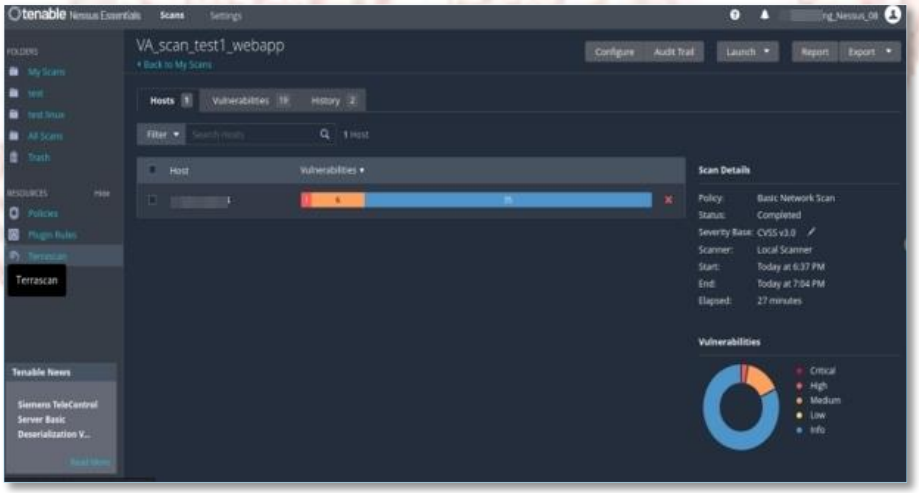
ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
21	NVT: Joomla! <= 3.9.19 Multiple Vulnerabilities	CVE-2020-15695: การขาดการตรวจสอบโทเค็นในส่วนขอการลบข้อมูลของ com_privacy ส่งผลให้เกิดช่องโหว่ CSRF CVE-2020-15696: การกรองข้อมูลเข้าและการ escape ไม่เพียงพอใน mod_random_image เปิดช่องให้โจมตีด้วย XSS CVE-2020-15697: ช่องข้อมูลที่ควรเป็นแบบอ่านอย่างเดียวในตารางผู้ใช้ (User Table) สามารถถูกแก้ไขโดยผู้ใช้ทั่วไปได้ CVE-2020-15698: การกรองข้อมูลที่ไม่เพียงพอในหน้าจอแสดงข้อมูลระบบ อาจทำให้ข้อมูลรับรอง Redis หรือ Proxy ถูกเปิดเผย CVE-2020-15699: ขาดการตรวจสอบความถูกต้องของตาราง usergroups อาจทำให้การตั้งค่าไซต์เสียหาย และCVE-2020-15700: การขาดการตรวจสอบโทเค็นใน endpoint ajax_install ของ com_installer ส่งผลให้เกิดช่องโหว่ CSRF
22	NVT: Joomla! 3.0.0 - 3.9.25 Multiple Vulnerabilities	CVE-2021-26030: การ escape ข้อมูลที่ไม่เพียงพอเปิดช่องให้โจมตีแบบ Cross-Site Scripting (XSS) โดยอาศัยพารามิเตอร์โลโก้ในเทมเพลตเริ่มต้นบนหน้าข้อผิดพลาด และCVE-2021-26031: การกรองข้อมูลไม่เพียงพอในค่าการตั้งค่าเลย์เอาต์ของโมดูล อาจนำไปสู่การเรียกใช้ไฟล์ภายในระบบโดยไม่ได้รับอนุญาต (Local File Inclusion - LFI)
23	NVT: Joomla! Multiple Vulnerabilities (20240802, 20240805)	CVE-2024-27185: ช่องโหว่การโจมตีแบบ Cache Poisoning ในระบบแบ่งหน้า อาจนำไปสู่การแสดงข้อมูลที่ไม่ถูกต้องหรือถูกควบคุมโดยผู้โจมตี และ CVE-2024-40743: ช่องโหว่ XSS ที่เกิดขึ้นในเมธอด strip* ของคลาส OutputFilter ซึ่งอาจเปิดทางให้มีการฝังโค้ดอันตรายผ่านข้อมูลที่ไม่ถูกกรองอย่างเหมาะสม

ตารางที่ 4-9 (ต่อ)

ลำดับ	ช่องโหว่ที่พบ	รายละเอียด
24	NVT: Joomla! 3.0.0 - 3.9.26 Multiple Vulnerabilities	CVE-2021-26032: นามสกุล .html ไม่ได้ถูกรวมอยู่ในรายการบล็อกไฟล์ที่สามารถประมวลผลได้ของ MediaHelper::canUpload ส่งผลให้เกิดช่องทางการโจมตีแบบ XSS CVE-2021-26033: การขาดการตรวจสอบโทเค็นใน endpoint สำหรับจัดเรียงข้อมูลผ่าน AJAX ทำให้เกิดช่องโหว่ CSRF และCVE-2021-26034: การขาดการตรวจสอบโทเค็นใน endpoint สำหรับดาวน์โหลดข้อมูลของ com_banners และ com_sysinfo ทำให้เกิดช่องโหว่ CSRF

4.1.2 ผลการสแกนด้วย Nessus ครั้งที่ 1

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ Nessus ในการทดสอบการสแกนครั้งที่ 1 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 7 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีระดับความรุนแรงสูง 1 ช่องโหว่ และช่องโหว่ที่มีระดับความรุนแรงปานกลาง 6 ช่องโหว่ แสดงดังภาพที่ 4-2



ภาพที่ 4-2 ผลการสแกนของโปรแกรม Nessus (รอบที่ 1)

โดยระดับความรุนแรงของ CVSS v3.0 (Common Vulnerability Scoring System) ใช้เกณฑ์การให้คะแนนได้ 4 ระดับคือ

Critical (9.0-10) – มีผลกระทบร้ายแรง ควรแก้ไขทันที

High (7.0 – 8.9) – มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย

Medium (4.0 - 6.9) – มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี

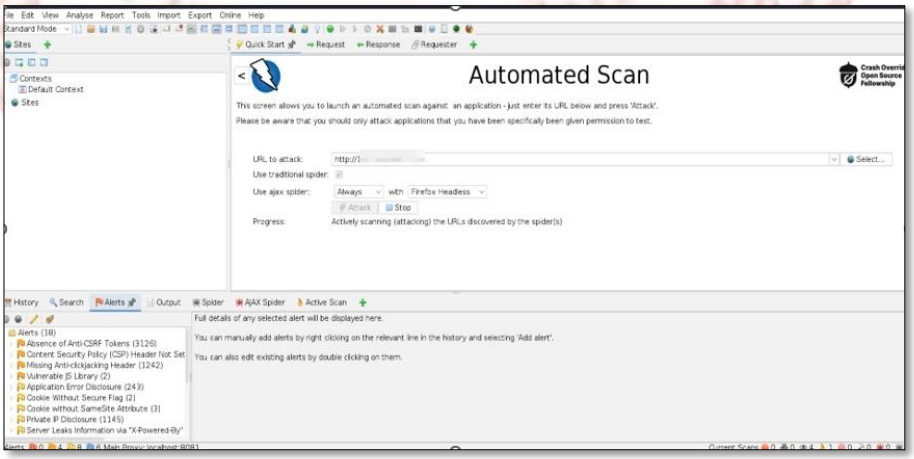
Low (0.0-3.9) – มีผลกระทบต่ำ มีความมั่นคงปลอดภัย ไม่สามารถโจมตีได้ง่าย

ตารางที่ 4-10 รายการช่องโหว่ที่ตรวจพบจากเครื่องมือ Nessus และระดับความรุนแรง

ช่องโหว่	ความรุนแรง
SSL Medium Strength Cipher Suites Supported (SWEET32)	สูง (High)
HTTP TRACE / TRACK Methods Allowed	ปานกลาง (Medium)
HTTP TRACE / TRACK Methods Allowed	ปานกลาง (Medium)
JQuery 1.2 < 3.5.0 Multiple XSS	ปานกลาง (Medium)
SSL Certificate Cannot Be Trusted	ปานกลาง (Medium)
TLS Version 1.0 Protocol Detection	ปานกลาง (Medium)
TLS Version 1.1 Deprecated Protocol	ปานกลาง (Medium)

4.1.3 ผลการสแกนด้วย OWASP ZAP ครั้งที่ 1

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ OWASP ZAP ในการทดสอบการสแกนครั้งที่ 1 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 12 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีระดับความรุนแรงปานกลาง 4 ช่องโหว่ และช่องโหว่ที่มีระดับความรุนแรงต่ำ 8 ช่องโหว่ แสดงดังภาพที่ 4-3



ภาพที่ 4-3 ผลการสแกนของโปรแกรม OWASP ZAP (รอบที่ 1)

ตารางที่ 4-11 รายการช่องโหว่ที่ตรวจพบจากเครื่องมือ OWASP ZAP และระดับความรุนแรง

ช่องโหว่	ความรุนแรง
Absence of Anti-CSRF Tokens	ปานกลาง (Medium)
Content Security Policy (CSP) Header Not Set	ปานกลาง (Medium)
Missing Anti-clickjacking Header	ปานกลาง (Medium)
Vulnerable JS Library	ปานกลาง (Medium)
Application Error Disclosure	ต่ำ (Low)
Cookie Without Secure Flag	ต่ำ (Low)
Cookie without SameSite Attribute	ต่ำ (Low)
Private IP Disclosure	ต่ำ (Low)
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	ต่ำ (Low)
Strict-Transport-Security Header Not Set	ต่ำ (Low)
Timestamp Disclosure - Unix	ต่ำ (Low)
X-Content-Type-Options Header Missing	ต่ำ (Low)

จากผลการทดสอบสแกนช่องโหว่จากทั้ง 3 เครื่องมือ สามารถสรุปผลการสแกนช่องโหว่ที่ตรวจพบ ดังแสดงในตารางที่ 4-12

ตารางที่ 4-12 จำนวนช่องโหว่ที่ตรวจพบ จาก 3 เครื่องมือ

เครื่องมือ	จำนวนช่องโหว่ที่พบทั้งหมด	Critical	High	Medium	Low
OpenVAS	72	-	33	39	-
Nessus	7	-	1	6	-
OWASP ZAP	12	-	-	4	8

4.2 รายละเอียดช่องโหว่ที่ตรวจพบแยกตามประเภทของช่องโหว่

จากช่องโหว่ที่พบจากทั้ง 3 เครื่องมือ สามารถสรุปประเภทช่องโหว่และวิธีการแก้ไขแต่ละช่องโหว่ได้ ดังนี้

4.2.1 ช่องโหว่ประเภท Injection Attacks

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

ช่องโหว่นี้เกี่ยวข้องกับ SQL Injection ซึ่งผู้โจมตีสามารถใช้เพื่อส่งคำสั่ง SQL ที่ไม่ได้รับอนุญาตเข้าสู่ฐานข้อมูลผ่านทางอินพุตที่ไม่ได้รับการตรวจสอบอย่างถูกต้องเหมาะสม ผู้โจมตีสามารถใช้ช่องโหว่นี้เข้าถึงข้อมูลสำคัญในฐานข้อมูล ความเสี่ยงต่อการรันคำสั่ง SQL ที่ไม่ปลอดภัย หากการโจมตีด้วยเทคนิคนี้สำเร็จ จะทำให้ผู้ไม่หวังดีเข้าถึง แก้ไขเปลี่ยนแปลง หรือลบข้อมูลในฐานข้อมูลได้ หรือ สั่งคำสั่งปิดฐานข้อมูล

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด และเพิ่มการป้องกันการโดยการกรองข้อมูลที่นำเข้า

4.2.2 ช่องโหว่ประเภท Directory and File Access Vulnerabilities

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

ช่องโหว่ที่เกี่ยวข้องกับการอนุญาตให้ผู้โจมตีเข้าถึงไฟล์ในระบบโดยไม่ได้รับอนุญาต ผู้ไม่หวังดีสามารถเรียกชื่อไฟล์หรือข้อมูลที่เก็บบนเครื่องบริการเว็บได้โดยตรง ผ่าน External Parameter เช่น ใส่คำสั่ง ?file=../../secret.txt ต่อท้าย URL ของเว็บไซต์ ก็จะเป็นการระบุชื่อไฟล์ที่ต้องการพร้อม Path ไฟล์ ทำให้ผู้ไม่ประสงค์ดี สามารถเข้าถึงไฟล์ที่สำคัญโดยไม่ได้รับอนุญาตได้

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด และมีการแก้ไข LFI หรือใช้งานฟีเจอร์การตรวจสอบไฟล์เซิร์ฟเวอร์

4.2.3 ช่องโหว่ประเภท Cross-Site and Browser-Based Attacks

เครื่องมือที่ตรวจพบ: OpenVAS, Nessus และ OWAPS ZAP

รายละเอียดช่องโหว่:

ช่องโหว่ที่เกี่ยวข้องกับการจัดการข้อมูลและไม่มีการคัดกรองที่เพียงพอ ที่ได้รับจาก

ผู้ใช้บริการว่าเป็นข้อมูลที่เชื่อถือได้หรือไม่ ทำให้ผู้ไม่ประสงค์ดีสามารถแทรกคำสั่งต่าง ๆ เข้าไปในเว็บเพจ เมื่อผู้ใช้บริการเรียกหน้าเว็บเพจนั้น ก็อาจจะถูกขโมยข้อมูลสำคัญไปได้ ซึ่งผู้ไม่ประสงค์ดีอาจจะสวมรอยและเข้าสู่ระบบไปยังเว็บไซต์เหมือนกับว่าเป็นผู้ใช้บริการตัวจริง

คำแนะนำในการแก้ไข: อัปเดต Joomla! เป็นเวอร์ชันล่าสุด อัปเกรดเป็น jQuery เวอร์ชัน 3.5.0 หรือเวอร์ชันที่ใหม่กว่า และเพิ่ม Anti-CSRF Tokens

4.2.4 ช่องโหว่ประเภท Access Control Vulnerabilities

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

ช่องโหว่ที่เกี่ยวข้องกับการควบคุมสิทธิ์การเข้าถึง เช่น Unauthorized Access ข้อมูลที่สำคัญอาจถูกแก้ไขโดยไม่ได้รับอนุญาต ข้อมูลสำคัญอาจถูกเปิดเผยหรือการเปลี่ยนแปลงการตั้งค่าของระบบและการตรวจสอบฟิลด์การค้นหาป้ายกำกับ (tags search fields) ที่ไม่เพียงพอภายในระบบ

คำแนะนำในการแก้ไข: อัปเดต Joomla! เป็นเวอร์ชันล่าสุด

4.2.5 ช่องโหว่ประเภท Validation Vulnerabilities

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

การปล่อยให้ input ที่อยู่นอกเหนือการคาดการณ์เข้ามาและสามารถรันคำสั่ง หรือแสดงผลที่ผิดปกติ โดยไม่ได้มีการตรวจสอบข้อมูลก่อน การตรวจสอบที่ไม่เพียงพอใน com_contact การตั้งค่า text filter ที่ไม่สอดคล้องกัน และไม่สามารถจัดการพารามิเตอร์ที่ป้อนเข้ามาได้อย่างเหมาะสม

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด,กฎ DPI Rule ,อัปเกรดเป็นระบบปฏิบัติการ Windows ตรวจสอบและกำหนดค่า Global Text filter Configuration ให้บล็อก HTML อินพุตสำหรับผู้ใช้ Guest ตั้งค่าให้ผู้ใช้ที่ไม่ได้รับอนุญาตไม่สามารถส่งโค้ด HTML หรือ JavaScript ได้ตรวจสอบความถูกต้องของ URL และพารามิเตอร์ที่รับเข้ามา และปิดการใช้งาน Shell Execute หากไม่จำเป็น ควรตรวจสอบการใช้งานและพิจารณาปิดฟังก์ชันที่ไม่ได้ใช้งาน

4.2.6 ช่องโหว่ประเภท Cryptographic and Session Issues

เครื่องมือที่ตรวจพบ: OpenVAS, Nessus และ OWAPS ZAP

รายละเอียดช่องโหว่:

มีการใช้โปรโตคอล TLSv1.0 และ TLSv1.1 ซึ่งเป็นโปรโตคอลที่มีความมั่นคงปลอดภัยต่ำ ช่องโหว่ที่เกี่ยวข้องกับการเข้ารหัสหรือจัดการ session เช่น Weak Ciphers ใบรับรองหมดอายุ และใบรับรอง X.509 ของเซิร์ฟเวอร์ไม่สามารถเชื่อถือได้ และการตั้งค่าคุกกี้ โดยไม่มีการกำหนด Secure Flag

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด, อัปเกรดซอฟต์แวร์ที่เกี่ยวข้องให้รองรับ Cipher Suites ที่ปลอดภัยกว่า ปิดใช้งาน TLS 1.0,1.1 ซึ่งเป็นเวอร์ชันต่ำ และเปิดใช้งาน TLS 1.2 หรือ TLS 1.3 ซึ่งเป็นเวอร์ชันที่สูงกว่า ตรวจสอบและแก้ไขวันหมดอายุของใบรับรอง ตั้งค่าแอตทริบิวต์ Secure สำหรับคุกกี้ที่ส่งผ่านการเชื่อมต่อ SSL/TLS และตรวจสอบให้แน่ใจว่ามีการกำหนด Same Site attribute ให้กับคุกกี้ทุกตัว

4.2.7 ช่องโหว่ประเภท Remote Execution

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

ช่องโหว่ประเภท Remote Code Execution ซึ่งทำให้ผู้โจมตีสามารถรันโค้ดอันตรายบนระบบได้ และสามารถรันโค้ดระยะไกลบนเซิร์ฟเวอร์ได้

คำแนะนำในการแก้ไข: อัปเดต Joomla! เป็นเวอร์ชันล่าสุด และใช้ไฟร์วอลล์และเครื่องมือป้องกัน RCE เพื่อป้องกันการโจมตี

4.2.8 ช่องโหว่ประเภท Information Disclosure

เครื่องมือที่ตรวจพบ: OpenVAS และ OWAPS ZAP

รายละเอียดช่องโหว่:

ช่องโหว่นี้เกิดจากการตรวจสอบสิทธิ์ที่ไม่เพียงพอ ทำให้ผู้โจมตีระยะไกลสามารถเข้าถึงข้อมูลที่สำคัญได้ ช่องโหว่ประเภท Path Disclosure ซึ่งช่วยให้ผู้โจมตีทราบโครงสร้างไดเรกทอรีของเซิร์ฟเวอร์ได้ มีข้อความแจ้งข้อผิดพลาด/คำเตือนที่อาจเปิดเผยข้อมูลที่ละเอียดอ่อน พบ Private IP หรือชื่อโฮสต์ภายใน Amazon EC2 ในการตอบกลับ HTTP Response Body และแอปพลิเคชันหรือเว็บเซิร์ฟเวอร์ได้เปิดเผยข้อมูล Timestamp ซึ่งอยู่ในรูปแบบ Unix

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด ลบ Private IP Address ออกจาก HTTP Response Body ตรวจสอบ เว็บเซิร์ฟเวอร์และแอปพลิเคชันเซิร์ฟเวอร์ ถูกตั้งค่าให้ปิดการแสดง X-Powered-By Header และหลีกเลี่ยงการเปิดเผย Timestamp ที่ไม่จำเป็น

4.2.9 ช่องโหว่ประเภท Header Issues

เครื่องมือที่ตรวจพบ: OpenVAS, Nessus และ OWAPS ZAP

รายละเอียดช่องโหว่:

เซิร์ฟเวอร์สนับสนุนวิธีการ TRACE/ TRACK ซึ่งอาจถูกใช้ในการโจมตีแบบ Cross-Site-Tracing (XST) และเว็บเบราว์เซอร์สมัยใหม่รองรับการใช้งาน Content-Security-Policy และ X-Frame-Options ผ่าน HTTP Header

คำแนะนำในการแก้ไข: ปิดการใช้งาน HTTP methods TRACE และ TRACK กำหนดค่า Content-Security-Policy (CSP) Header และตรวจสอบว่าเว็บเซิร์ฟเวอร์หรือแอปพลิเคชันมีการตั้งค่า X-Content-Type-Options เป็น 'nosniff' สำหรับทุก Response

4.2.10 ช่องโหว่ประเภท Libraries and Frameworks

เครื่องมือที่ตรวจพบ: OWAPS ZAP

รายละเอียดช่องโหว่:

ไลบรารีที่ตรวจพบ ExampleLibrary เวอร์ชัน x.y.z มีช่องโหว่ด้านความมั่นคงปลอดภัย และไลบรารีหรือซอฟต์แวร์เวอร์ชันเก่า มีความมั่นคงปลอดภัยในระดับที่ต่ำ เสี่ยงต่อการถูกโจมตี

คำแนะนำในการแก้ไข: อัปเกรดเป็นเวอร์ชันล่าสุดของ ExampleLibrary

4.2.11 ช่องโหว่ประเภท Multiple Vulnerabilities

เครื่องมือที่ตรวจพบ: OpenVAS

รายละเอียดช่องโหว่:

Joomla! ที่ติดตั้งบนเว็บเซิร์ฟเวอร์มีเวอร์ชันต่ำ ซึ่งทำให้ระบบได้รับผลกระทบหลายด้าน เช่น Malicious File Upload ช่องโหว่ที่อนุญาตให้ผู้โจมตีอัปโหลดไฟล์อันตราย เข้าสู่ระบบ Stored Cross-Site Scripting (XSS) ช่องโหว่ที่อนุญาตให้ผู้โจมตีแทรกโค้ดอันตราย ลงในระบบและรันในเบราว์เซอร์ของผู้ใช้

คำแนะนำในการแก้ไข: อัปเดต Joomla เป็นเวอร์ชันล่าสุด

4.3 ผลการประเมินและการวิเคราะห์ความเสี่ยงจากช่องโหว่โดยการจัดกลุ่มตาม OWASP Top 10 2021

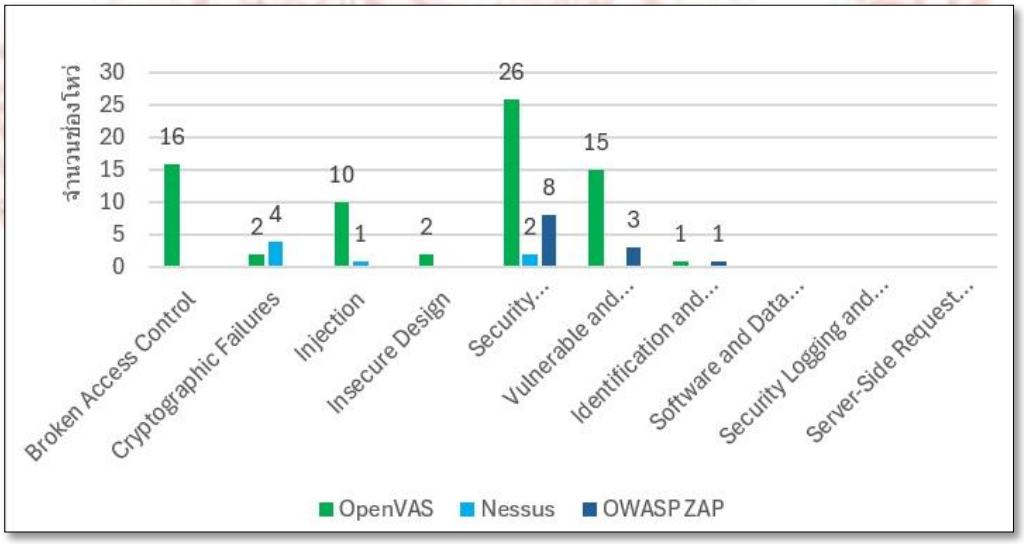
จากข้อมูลการประเมินช่องโหว่และการวิเคราะห์ความเสี่ยงจากช่องโหว่โดยการจัดกลุ่มตาม OWASP Top 10 2021 ที่ตรวจพบโดย OpenVAS, Nessus และ OWASP ZAP พบว่า

A05: Security Misconfiguration เป็นช่องโหว่ที่พบมากที่สุด 36 ช่องโหว่ ตรวจพบได้ทั้ง 3 เครื่องมือ เป็นการตั้งค่าระบบความมั่นคงปลอดภัยที่ผิดพลาด หรือการจำกัดสิทธิ์การเข้าถึงระบบ

รองลงมา คือ A06: Vulnerable and Outdated Components 18 ช่องโหว่ คือการใช้ซอฟต์แวร์ที่เก่าหรือล้าสมัยยังไม่ได้อัปเดตให้เป็นปัจจุบัน แสดงดังตารางที่ 4-13 และภาพที่ 4-4

ตารางที่ 4-13 จำนวนช่องโหว่จากการวิเคราะห์ความเสี่ยงโดยการจัดกลุ่มตาม OWASP Top 10 2021

OWASP Top 10	OpenVAS	Nessus	OWASP ZAP	Total
Broken Access Control	16	-	-	16
Cryptographic Failures	2	4	-	6
Injection	10	1	-	11
Insecure Design	2	-	-	2
Security Misconfiguration	26	2	8	36
Vulnerable and Outdated Components	15	-	3	18
Identification and Authentication Failures	1	-	1	2
Software and Data Integrity Failures	-	-	-	-
Security Logging and Monitoring Failures	-	-	-	-
Server-Side Request Forgery (SSRF)	-	-	-	-



ภาพที่ 4-4 จำนวนช่องโหว่จากการวิเคราะห์ความเสี่ยงโดยการจัดกลุ่มตาม OWASP Top 10 2021

รายละเอียดของช่องโหว่แต่ละกลุ่มที่พบในแต่ละเครื่องมือ แสดงดังตารางที่ 4-14

ตารางที่ 4-14 รายการวิเคราะห์ความเสี่ยงโดยการจัดกลุ่มตาม OWASP Top 10 2021

OWASP Top 10	ช่องโหว่ที่พบ	Severity
A01: Broken Access Control	**OpenVAS**	
	- Joomla! 3.0.0 - 3.9.14 Multiple Vulnerabilities	High
	- Joomla! 3.2.0 - 3.9.15 CSRF Vulnerability	High
	- Joomla! 3.7.0 - 3.9.18 CSRF Vulnerability	High
	- Joomla! 3.0.0 - 3.9.20 Directory Traversal Vulnerability	Medium
	- Joomla! < 3.9.11 Mail Submission Vulnerability	Medium
	- Joomla! 1.6.0 - 4.4.0, 5.0.0 Information Disclosure Vulnerability	High
	- Joomla! 3.8.0 - 3.9.13 Path Disclosure Vulnerability	Medium
	- Joomla 'Unpublished Tags' Information Disclosure Vulnerability	Medium
	- Joomla! 3.7.0 - 3.9.15 Access Control Vulnerability	High
	- Joomla! < 3.8.13 ACL Violation Vulnerability	High
	- Joomla < 3.8.12 ACL Violation Vulnerability	High
	- Joomla! 2.5.0 - 3.9.15 Access Control Vulnerability	High
	- Joomla! 1.7.0 - 3.9.22 ACL Violation Vulnerability	High
	- Joomla! 3.0.0 - 3.9.23 ACL Vulnerability	Medium
	- Joomla! 1.6.0 - 3.9.24 ACL Vulnerability	Medium
	- Joomla! < 3.8.13 Access Level Violation Vulnerability	Medium
A02: Cryptographic Failures	**OpenVAS**	
	- SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	High
	- SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	Medium
	Nessus	
	-SSL Medium Strength Cipher Suites Supported (SWEET32)	High
	- SSL Certificate Cannot Be Trusted	Medium
	- TLS Version 1.0 Protocol Detection	Medium
	- TLS Version 1.1 Deprecated Protocol	Medium

ตารางที่ 4-14 (ต่อ)

OWASP Top 10	ช่องโหว่ที่พบ	Severity
A03: Injection	**OpenVAS**	
	- Joomla! 3.0.0 - 3.9.22 SQL Injection Vulnerability	Critical
	- Joomla! 2.5.0 - 3.9.13 SQL Injection Vulnerability	Critical
	- Joomla! 1.7.0 - 3.9.15 SQL Injection Vulnerability	Critical
	- Joomla 'User Notes list view' SQL Injection Vulnerability	High
	- Joomla! 2.5.0 - 3.9.18 Text Filter Vulnerability	High
	- Joomla 'com_elds' RCE Vulnerability (20180506)	Medium
	- Joomla! <= 3.9.19 Multiple Vulnerabilities	Medium
	- Joomla! 3.0.0 - 3.9.27 Multiple XSS Vulnerabilities	Medium
	- Joomla! 2.5.0 - 3.9.24 Multiple XSS Vulnerabilities	Medium
	- Joomla! URL Validation Vulnerability (20240801)	Medium
	Nessus	
	- JQuery 1.2 < 3.5.0 Multiple XSS	Medium
A04: Insecure Design	**OpenVAS**	
	- Joomla! Open Redirect Vulnerability (20240202)	Medium
	- Joomla! 3.0.0 - 3.9.20 Open Redirect Vulnerability	Medium
A05: Security Misconfiguration	**OpenVAS**	
	- Joomla Multiple Vulnerabilities (20180507, 20180505, 20180504)	High
	- Joomla! 3.0.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	High
	- Joomla! 3.7.0 <= 3.8.3 Multiple Vulnerabilities	High
	- Joomla! Core 'PHP' Local File Inclusion Vulnerability (20180601)	High
	- Joomla! < 3.8.13 RCE Vulnerability	Medium
	- Joomla! XSS Vulnerability (20240203)	Medium
	- Joomla! Multiple Vulnerabilities (20240802, 20240805)	Medium
	- Joomla! XSS Vulnerability (20240205)	Medium
	- Joomla! < 3.9.12 XSS Vulnerability	Medium

ตารางที่ 4-14 (ต่อ)

OWASP Top 10	ช่องโหว่ที่พบ	Severity
	- Joomla! 1.7.0 <= 3.9.5 XSS Vulnerability	Medium
	- Joomla! 3.0.0 - 3.9.15 Multiple Vulnerabilities	Medium
	- Joomla! 3.0.0 - 3.9.18 Multiple XSS Vulnerabilities	Medium
	- Joomla! 3.0.0 - 3.9.25 Multiple Vulnerabilities	Medium
	- Joomla! 3.1.0 - 3.9.23 XSS Vulnerability	Medium
	- Joomla! 3.7.0 - 3.10.6 XSS Vulnerability	Medium
	- Joomla 'Chromes' module XSS Vulnerability	Medium
	- Joomla! 'Language Switcher' Module XSS Vulnerability (20180602)	Medium
	- Joomla! 'Media Manager' XSS Vulnerability (20180509)	Medium
	- Joomla! Multiple XSS Vulnerabilities (20240703, 20240704)	Medium
	- Joomla 'Uri' class XSS Vulnerability	Medium
	- Joomla! XSS Vulnerability (20240705)	Medium
	- HTTP Debugging Methods (TRACE/TRACK) Enabled	Medium
	- Joomla 'Redirect' Method XSS Vulnerability (20180508)	Medium
	- HTTP Debugging Methods (TRACE/TRACK) Enabled	Medium
	- Missing 'Secure' Cookie Attribute (HTTP)	Medium
	Nessus	
	- HTTP TRACE / TRACK Methods Allowed	Medium
	- HTTP TRACE / TRACK Methods Allowed	Medium
	OWASP ZAP	
	- Content Security Policy (CSP) Header Not Set	Medium
	- Missing Anti-clickjacking Header	Medium
	- Application Error Disclosure	Low
	- Private IP Disclosure	Low
	- Strict-Transport-Security Header Not Set	Low
	- X-Content-Type-Options Header Missing	Low
	- Cookie Without Secure Flag	Low
	- Cookie without SameSite Attribute	Low

ตารางที่ 4-14 (ต่อ)

OWASP Top 10	ช่องโหว่ที่พบ	Severity
A06: Vulnerable and Outdated Components	**OpenVAS**	
	- Joomla! < 3.9.7 Multiple Vulnerabilities	High
	- Joomla! 2.5.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	High
	- Joomla! < 3.9.3 Multiple Vulnerabilities	High
	- Joomla < 3.8.12 Multiple Vulnerabilities	High
	- Joomla! < 3.9.5 Multiple Vulnerabilities	High
	- Joomla! 3.0.0 - 3.9.24 Multiple Vulnerabilities	High
	- Joomla! 3.0.0 - 3.9.24 Multiple Vulnerabilities	High
	- Joomla! < 3.8.13 Multiple Vulnerabilities	High
	- Joomla! < 3.9.13 Multiple Vulnerabilities	High
	- Joomla! < 3.9.4 Multiple Vulnerabilities	High
	- Joomla! 2.5.0 - 3.9.22 Multiple Vulnerabilities	High
	- Joomla! 3.0.0 - 3.9.26 Multiple Vulnerabilities	Medium
	- Joomla! 3.0.0 - 3.9.27 Multiple XSS Vulnerabilities	Medium
	- Joomla! < 3.9.2 Multiple Vulnerabilities	Medium
	- Joomla! 2.5.0 - 3.9.16 Multiple Vulnerabilities	Medium
	OWASP ZAP	
	- Vulnerable JS Library	Medium
	- Server Leaks Information via "X-Powered-By" HTTP Header	Low
	- Timestamp Disclosure – Unix	Low
A07: Identification & Authentication Failures	**OpenVAS**	
	- Joomla! Session Expiration Vulnerability (20240201)	Medium
	OWASP ZAP	
	- Absence of Anti-CSRF Tokens	Medium
A08: Software & Data Integrity Failures	-	
A09: Security Logging and Monitoring Failures	-	
A10: SSRF	-	

4.4 การแก้ไขช่องโหว่

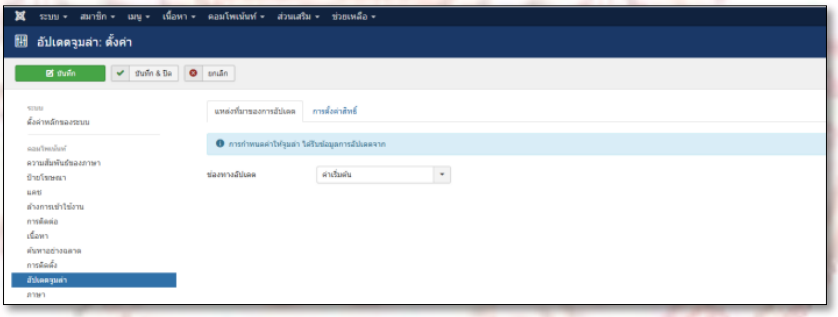
ในการบริหารจัดการความมั่นคงปลอดภัยของระบบ การแก้ไขช่องโหว่ที่มีความเสี่ยงระดับวิกฤตและช่องโหว่ที่มีความเสี่ยงระดับสูง ควรให้ความสำคัญเป็นลำดับแรก และจะต้องรีบดำเนินแก้ไขอย่างเร่งด่วน เพื่อป้องกันการโจมตีทางไซเบอร์ และลดผลกระทบต่อองค์กร เนื่องจากช่องโหว่เหล่านี้ผู้ไม่ประสงค์ดีใช้ช่องโหว่เพื่อ เข้าถึงข้อมูลสำคัญหรือทำให้ระบบล่ม ซึ่งอาจส่งผลต่อความมั่นคงขององค์กรและความมั่นคงปลอดภัยของข้อมูล

4.4.1 การแก้ไขช่องโหว่ที่ที่ตรวจพบจากเครื่องมือ OpenVAS

จากการวิเคราะห์ข้อมูลช่องโหว่ที่ตรวจพบโดย OpenVAS พบว่า เครื่องมือ OpenVAS มีช่องโหว่ที่มีความเสี่ยงสูง (High) 33 ช่องโหว่และมีความเชื่อมโยงกับช่องโหว่ที่มีความเสี่ยงปานกลางอีก 39 ช่องโหว่ ดังนั้นจึงมีความจำเป็นต้องดำเนินการแก้ไขเร่งด่วน ช่องโหว่ที่ตรวจพบส่วนใหญ่จะเป็นเวอร์ชันที่ล้าสมัยจึงต้องทำการอัปเดตเพื่อปิดช่องโหว่ โดยมีการแก้ไขดังนี้

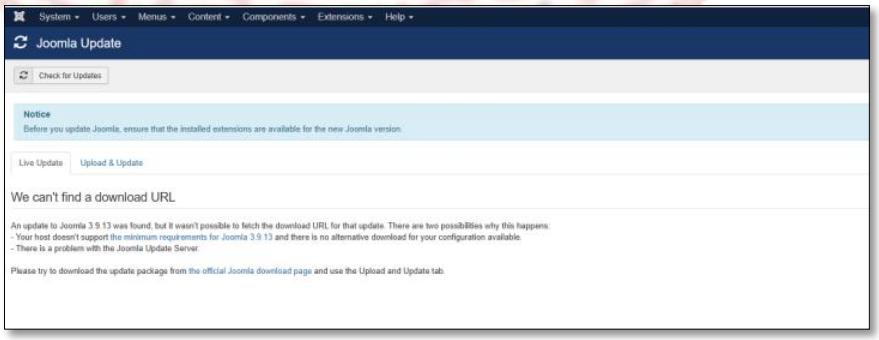
Login Joomla โดยเข้าไปที่หน้าสำหรับผู้ดูแลระบบ และทำการ Login โดยใช้ชื่อบัญชีผู้ใช้งานที่มีสิทธิ์ในการแก้ไข จากนั้น

1. เลือกไปที่เมนู ตั้งค่า อัปเดต Joomla!



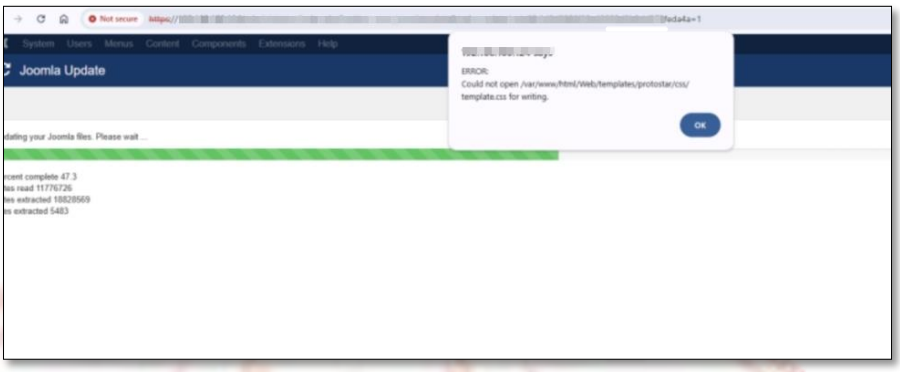
ภาพที่ 4-5 การตั้งค่าการอัปเดต Joomla!

2. เลือกไปที่เมนู Joomla! Update แล้วกด Check for update



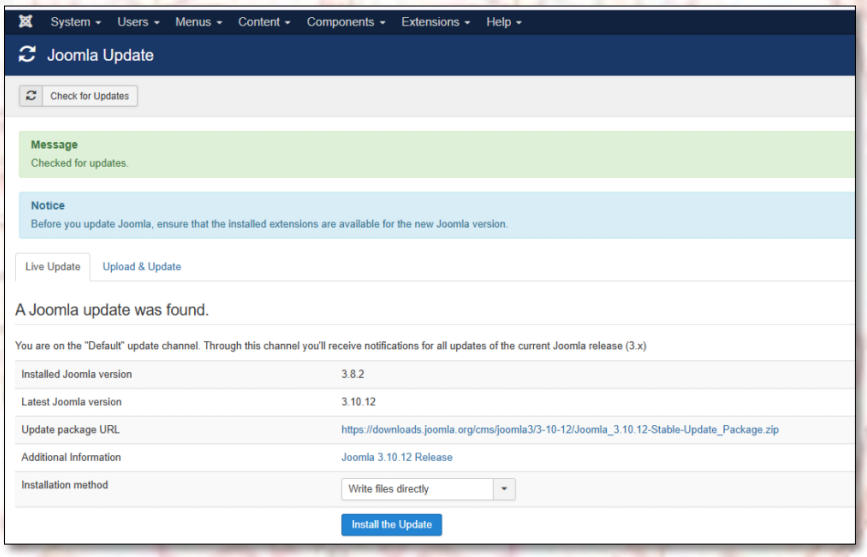
ภาพที่ 4-6 เมนูอัปเดต Joomla!

3. จะมีแถบความคืบหน้าแสดงว่ากำลังอัปเดตไฟล์ของ Joomla!



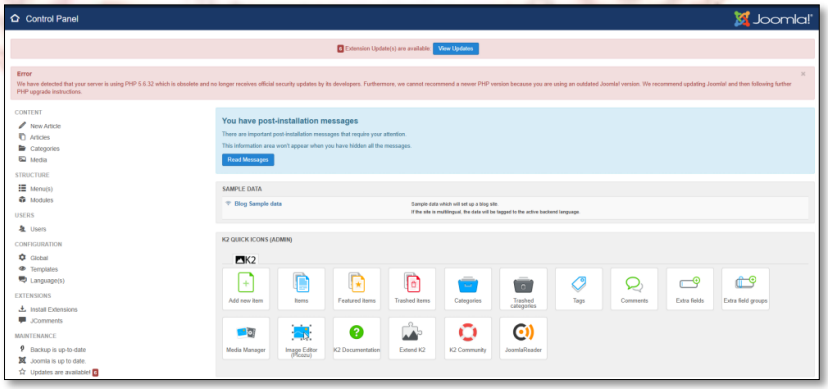
ภาพที่ 4-7 ความคืบหน้าการอัปเดต Joomla!

4. เมื่ออัปเดตสำเร็จ จะแสดงข้อความว่า Checked for update.



ภาพที่ 4-8 การอัปเดต Joomla! เสร็จสิ้น

5. หลังจากอัปเดตเรียบร้อยแล้ว จากนั้นทดสอบการใช้งาน



ภาพที่ 4-9 เวอร์ชั่นใหม่ของ Joomla!

จากการดำเนินการอัปเกรดเวอร์ชันที่สูงขึ้น ระบบแจ้งเตือน PHP เวอร์ชัน 5.6.32 ซึ่งล้าสมัยและไม่ได้รับการอัปเดตความมั่นคงปลอดภัยจากผู้พัฒนาอีกต่อไปแล้ว ไม่สามารถอัปเกรด PHP ไปยังเวอร์ชันใหม่กว่านี้ได้ เนื่องจากยังใช้งาน Joomla! เวอร์ชันเก่าอยู่

4.4.2 การแก้ไขช่องโหว่ที่ที่ตรวจพบจากเครื่องมือ Nessus

จากการวิเคราะห์ข้อมูลช่องโหว่ที่ตรวจพบโดย Nessus พบว่า เครื่องมือ Nessus มีช่องโหว่ที่มีความเสี่ยงสูง (High) 1 ช่องโหว่และมีความเชื่อมโยงกับช่องโหว่ที่มีความเสี่ยงปานกลางอีก 2 ช่องโหว่ ดังนั้นจึงมีความจำเป็นต้องดำเนินการแก้ไขเร่งด่วน สำหรับเครื่องมือนี้ โดยมีการแก้ไขดังนี้

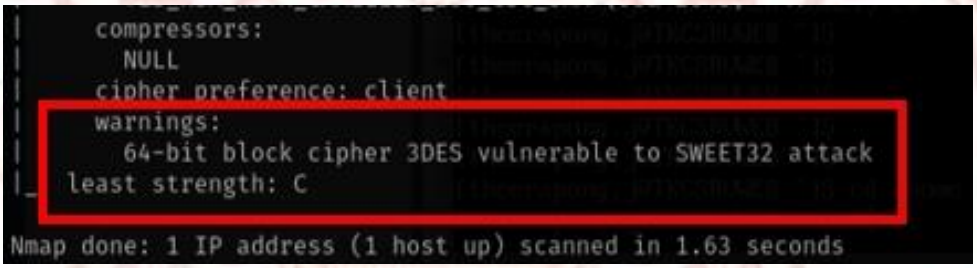
ชื่อช่องโหว่ : SSL Medium Strength Cipher Suites Supported (SWEET32)	
ระดับความรุนแรง	สูง (High)
รายละเอียดช่องโหว่	- ระบบที่ตรวจสอบรองรับการใช้ Cipher Suites ระดับกลางที่เข้ารหัสด้วยคีย์ความยาว 64 ถึง 112 บิต หรือ 3DES ซึ่งมีความมั่นคงปลอดภัยน้อยกว่าและอาจถูกโจมตีได้ง่ายหากผู้โจมตีอยู่ในเครือข่ายเดียวกัน - OpenSSL เวอร์ชันเก่า
OWASP Top Ten 2021	A02: Cryptographic Failures
คำแนะนำในการแก้ไข	- ปรับการตั้งค่าแอปพลิเคชันให้หยุดการใช้งาน Cipher Suites ระดับกลาง เช่น 3DES อัปเกรดซอฟต์แวร์ที่เกี่ยวข้องให้รองรับ Cipher Suites ที่มีความมั่งคงปลอดภัยกว่า - อัปเกรด OpenSSL เป็นเวอร์ชันใหม่กว่า เช่น OpenSSL 3.0 หรือ 1.1.1 เพื่อความมั่นคงปลอดภัยและการรองรับ TLS ล่าสุด

วิธีการแก้ไข:

1. ตรวจสอบการเข้าถึงโดยการพิมพ์คำสั่ง Ping< Ip Address ของเว็บไซต์ที่ต้องการทดสอบ> เพื่อทดสอบการเชื่อมต่อกับ Host

ภาพที่ 4-10 การทดสอบการเชื่อมต่อกับ Host

2. พิมพ์คำสั่ง nmap -sV –script ssl-enum-ciphers < Ip Address> จะได้ผลลัพท์ที่เป็นจุดอ่อนหรือช่องโหว่ดังนี้



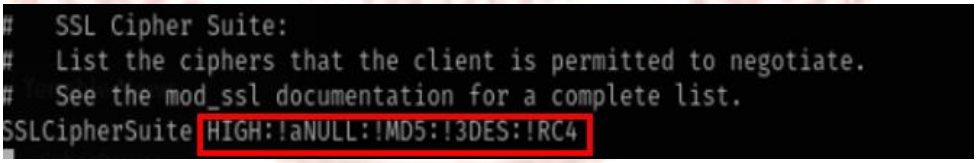
ภาพที่ 4-11 ผลที่ได้จากการพิมพ์คำสั่ง nmap -sV –script ssl-enum-ciphers

3. เข้าไปแก้ไขไฟล์ ssl.conf โดยการ พิมพ์คำสั่ง

[root@TKCSRVWEB theerapong.j]# vi /etc/httpd/conf.d/ssl.conf เพื่อเข้าถึงไฟล์ทำการแก้ไขไฟล์ โดยการกด I เพื่อแก้ไขข้อความ จาก

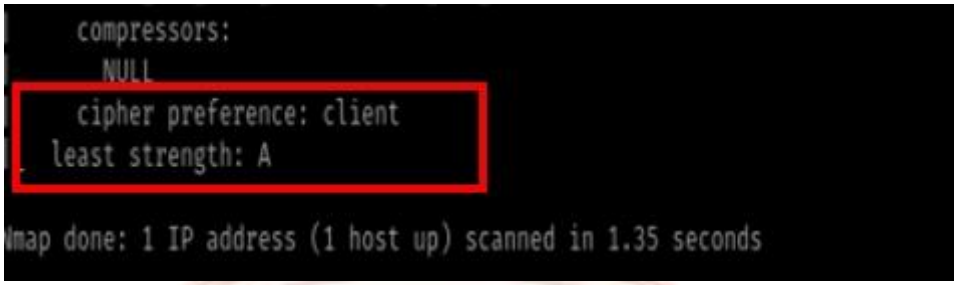
SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA เปลี่ยนเป็น

SSLCipherSuite HIGH:!aNULL:!MD5:!3DES:!RC4 และทำการบันทึก



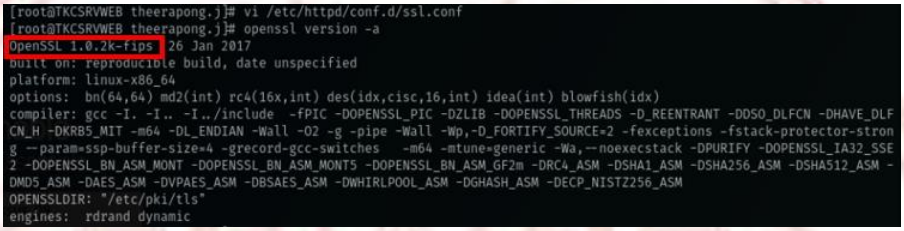
ภาพที่ 4-12 ข้อความที่ต้องทำการแก้ไขจากไฟล์ ssl.conf (SSLCipherSuite)

4. พิมพ์คำสั่ง nmap -sV –script ssl-enum-ciphers< Ip Address> เพื่อทำการตรวจสอบผลลัพท์อีกครั้ง



ภาพที่ 4-13 ผลลัพท์ที่ได้ทำการแก้ไขแล้ว

5. ทำการตรวจสอบ OpenSSL เวอร์ชันปัจจุบัน โดยการพิมพ์คำสั่ง openssl version -a



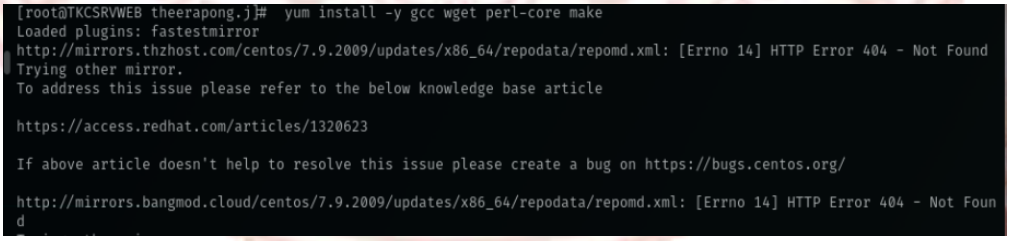
ภาพที่ 4-14 การตรวจสอบเวอร์ชันปัจจุบันของ OpenSSL

6. อัปเกรด OpenSSL โดยการติดตั้ง Dependencies

พิมพ์คำสั่ง yum groupinstall "Development Tools" -y

ภาพที่ 4-15 ผลลัพท์จาก คำสั่ง yum groupinstall "Development Tools" -y

7. พิมพ์คำสั่ง yum install -y gcc wget perl-core make

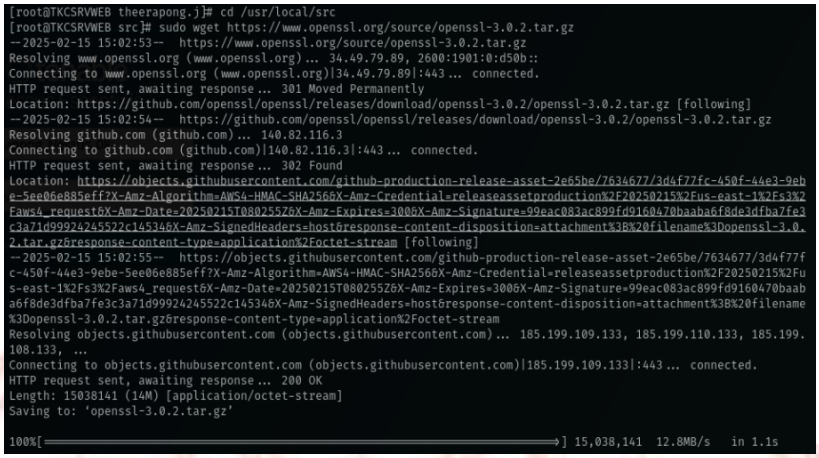


ภาพที่ 4-16 ผลลัพท์จาก คำสั่ง yum install -y gcc wget perl-core make

8. จากนั้นทำการดาวน์โหลดและติดตั้ง OpenSSL เวอรฺชันล่าสุด

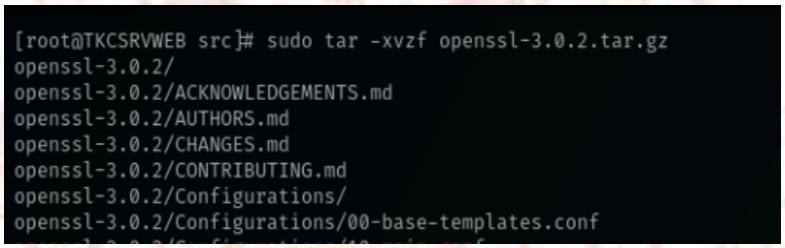
พิมพ์คำสั่ง cd /usr/local/src และ พิมพ์คำสั่ง sudo wget

https://www.openssl.org/source/openssl-3.0.2.tar.gz



ภาพที่ 4-17 การดาวน์โหลดและติดตั้ง OpenSSL เวอรฺชันล่าสุด

9. พิมพ์คำสั่ง sudo tar -xvzf openssl-3.0.2.tar.gz

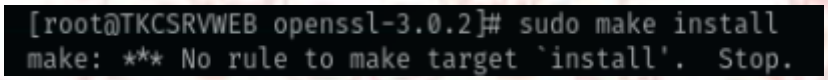


ภาพที่ 4-18 ผลลัพท์จากคำสั่ง sudo tar -xvzf openssl-3.0.2.tar.gz

10. พิมพ์คำสั่ง cd openssl-3.0.2 คอมไพล์ OpenSSLใหม่ พิมพ์คำสั่ง

sudo ./config --prefix=/usr/local/openssl --openssldir=/usr/local/openssl

พิมพ์คำสั่ง sudo make -j$(nproc) และ พิมพ์คำสั่ง sudo make install



ภาพที่ 4-19 ผลลัพท์จากคำสั่ง sudo make install

11. ทำการแทนที่ OpenSSL เวอร์ชันเก่า พิมพ์คำสั่ง

sudo mv /usr/bin/openssl /usr/bin/openssl.backup

sudo ln -s /usr/local/openssl/bin/openssl /usr/bin/openssl

และทำการตรวจสอบเวอร์ชันใหม่ พิมพ์คำสั่ง openssl version -a

ภาพที่ 4-20 ผลลัพท์จากคำสั่ง openssl version -a

ชื่อช่องโหว่ : TLS Version 1.0 Protocol Detection และ TLS Version 1.1 Deprecated Protocol	
ระดับความรุนแรง	ปานกลาง (Medium)
รายละเอียดช่องโหว่	- เซิร์ฟเวอร์รองรับโปรโตคอล TLS 1.0 ซึ่งมีช่องโหว่ด้านการเข้ารหัสและไม่ได้รับการสนับสนุนจากเบราว์เซอร์หลัก และบริการระยะไกลยอมรับการเชื่อมต่อที่เข้ารหัสด้วย TLS 1.1 ซึ่ง TLS 1.1 ไม่มีการรองรับ Cipher Suites ที่แนะนำในปัจจุบัน การเข้ารหัสที่รองรับก่อนการคำนวณ MAC (Message Authentication Code) - โหมดการเข้ารหัสที่มีการรับรองความถูกต้อง ตั้งแต่วันที่ 31 มีนาคม 2020 เป็นต้นไป จุดเชื่อมต่อ (Endpoints) ที่ไม่รองรับ TLS 1.2 หรือสูงกว่า จะไม่สามารถทำงานได้อย่างถูกต้องกับเบราว์เซอร์หลักและผู้ให้บริการรายใหญ่
OWASP Top Ten 2021	A02: Cryptographic Failures
คำแนะนำในการแก้ไข	ปิดใช้งาน TLS 1.0 ซึ่งเป็นเวอร์ชันต่ำ และเปิดใช้งาน TLS 1.2 หรือ TLS 1.3 ซึ่งเป็นเวอร์ชันที่สูงกว่า

วิธีการแก้ไข :

1. เข้าไปแก้ไขไฟล์ ssl.conf โดยการ พิมพ์คำสั่ง [root@TKCSRVWEB theerapong.j]# vi /etc/httpd/conf.d/ssl.conf เพื่อเข้าถึงไฟล์ทำการแก้ไขไฟล์ โดยการกด I เพื่อแก้ไขข้อความ จาก SSLProtocol All -SSLv2 -SSLv3 เปลี่ยนเป็น SSLProtocol +TLSv1.2 +TLSv1.3 และทำการบันทึก

```
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2 -SSLv3
```

ภาพที่ 4-21 ข้อความที่ต้องต้องทำการแก้ไขจากไฟล์ ssl.conf (SSLProtocaol)

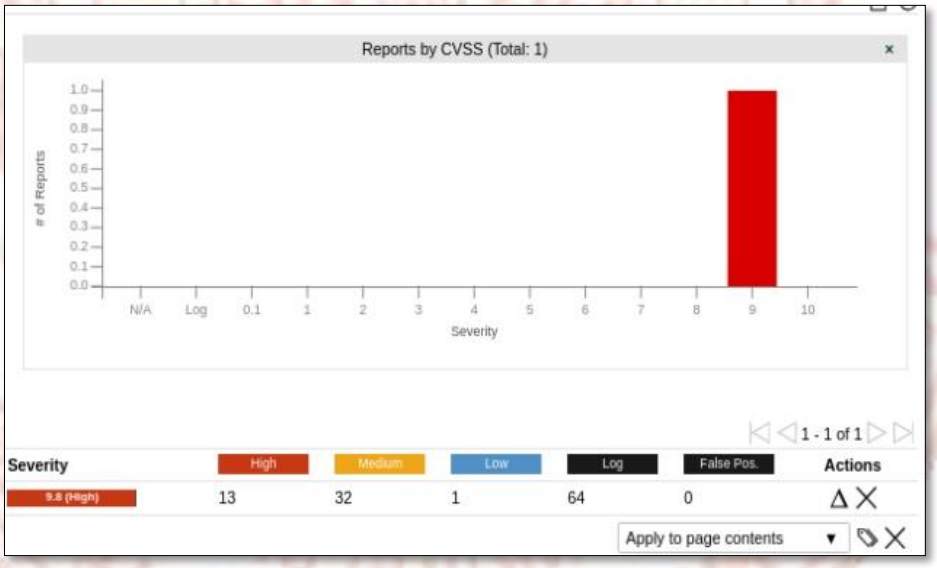
4.4.3 การแก้ไขช่องโหว่ที่ที่ตรวจพบจากเครื่องมือ OWASP ZAP

จากการวิเคราะห์ข้อมูลช่องโหว่ที่ตรวจพบโดย OWASP ZAP พบว่า เครื่องมือ OWASP ZAP ไม่มีช่องโหว่ที่มีระดับความรุนแรงสูง (High) ดังนั้นจึง ไม่มีความจำเป็นต้องดำเนินการแก้ไขเร่งด่วนสำหรับเครื่องมือนี้

4.5 ผลการสแกนช่องโหว่ครั้งที่ 2

4.5.1 ผลการสแกนด้วย OpenVAS ครั้งที่ 2

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ OpenVAS ในการทดสอบการสแกนครั้งที่ 2 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 13 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีระดับความรุนแรงสูง 33 ช่องโหว่ ช่องโหว่ที่มีระดับความรุนแรงปานกลาง 32 ช่องโหว่ และช่องโหว่ที่มีระดับความรุนแรงต่ำ 1 ช่องโหว่ แสดงดังภาพที่ 4-22



ภาพที่ 4-22 ผลการสแกนของโปรแกรม OpenVAS (รอบที่ 2)

โดยระดับความรุนแรงของ CVSS v2.0 (Common Vulnerability Scoring System) ใช้เกณฑ์การให้คะแนนได้ 3 ระดับคือ

High (7.0 – 10.0) – มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย

Medium (4.0 - 6.9) – มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี

Low (0.0-3.9) – มีผลกระทบต่ำ มีความมั่นคงปลอดภัย ไม่สามารถโจมตีได้ง่าย

ตารางที่ 4-15 รายการแก้ไขช่องโหว่ที่ตรวจพบจากเครื่องมือ OpenVAS (รอบที่2)

ช่องโหว่	ความรุนแรง	การแก้ไข
NVT: Joomla Multiple Vulnerabilities (20180507, 20180505, 20180504)	High	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.22 SQL Injection Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla 3.7.0 <= 3.8.3 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.7 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.10.6, 4.0.0 - 4.1.0 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.13 SQL Injection Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 1.7.0 - 3.9.15 SQL Injection Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.3 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla < 3.8.12 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla < 3.9.5 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 3.2.0 - 3.9.24 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! Core 'PHP' Local File Inclusion Vulnerability (20180601)	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.8.13 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.13 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.14 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla 'User Notes list view' SQL Injection Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.2.0 - 3.9.15 CSRF Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.7.0 - 3.9.15 Access Control Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.7.0 - 3.9.18 CSRF Vulnerability	High	ไม่พบช่องโหว่

ตารางที่ 4-15 (ต่อ)

ช่องโหว่	ความรุนแรง	การแก้ไข
NVT: Joomla Multiple Vulnerabilities (20180502, 20180501)	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.8.13 ACL Violation Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.24 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.27 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.18 Text filter Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla < 3.8.12 ACL Violation Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.15 Access Control Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 1.6.0 - 4.4.0, 5.0.0 Information Disclosure Vulnerability	High	ยังไม่ได้แก้ไข
NVT: Joomla! < 3.9.4 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	High	ยังไม่ได้แก้ไข
NVT: Joomla! 2.5.0 - 3.9.22 Multiple Vulnerabilities	High	ไม่พบช่องโหว่
NVT: Joomla! 1.7.0 - 3.9.22 ACL Violation Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! < 3.8.13 RCE Vulnerability	High	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.26 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla 'com_elds' RCE Vulnerability (20180506)	Medium	ไม่พบช่องโหว่
NVT: Joomla! XSS Vulnerability (20240203)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! Open Redirect Vulnerability (20240202)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! Multiple Vulnerabilities (20240802, 20240805)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! XSS Vulnerability (20240205)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! URL Validation Vulnerability (20240801)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! <= 3.9.19 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.12 XSS Vulnerability	Medium	ไม่พบช่องโหว่

ตารางที่ 4-15 (ต่อ)

ช่องโหว่	ความรุนแรง	การแก้ไข
NVT: Joomla! 1.7.0 <= 3.9.5 XSS Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.24 Multiple XSS Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.15 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.18 Multiple XSS Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.20 Open Redirect Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.25 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.27 Multiple XSS Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.1.0 - 3.9.23 XSS Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.7.0 - 3.10.6 XSS Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.2 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla 'Chromes' module XSS Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla 'Language Switcher' Module Cross Site Scripting Vulnerability (20180602)	Medium	ไม่พบช่องโหว่
NVT: Joomla 'Media Manager' XSS Vulnerability (20180509)	Medium	ไม่พบช่องโหว่
NVT: Joomla! Multiple XSS Vulnerabilities (20240703, 20240704)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla 'Uri' class XSS Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! XSS Vulnerability (20240705)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! 3.0.0 - 3.9.20 Directory Traversal Vulnerability	Medium	ไม่พบช่องโหว่
NVT: HTTP Debugging Methods (TRACE/TRACK) Enabled	Medium	ไม่พบช่องโหว่

ตารางที่ 4-15 (ต่อ)

ช่องโหว่	ความรุนแรง	การแก้ไข
NVT: Joomla! Session Expiration Vulnerability (20240201)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla! 3.8.0 - 3.9.13 Path Disclosure Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 3.0.0 - 3.9.23 ACL Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 2.5.0 - 3.9.16 Multiple Vulnerabilities	Medium	ไม่พบช่องโหว่
NVT: Joomla! < 3.9.11 Mail Submission Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! 1.6.0 - 3.9.24 ACL Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Missing 'Secure' Cookie Attribute (HTTP)	Medium	ยังไม่ได้แก้ไข
NVT: Joomla 'Redirect' Method XSS Vulnerability (20180508)	Medium	ไม่พบช่องโหว่
NVT: Joomla 'Unpublished Tags' Information Disclosure Vulnerability	Medium	ไม่พบช่องโหว่
NVT: Joomla! < 3.8.13 Access Level Violation Vulnerability	Medium	ไม่พบช่องโหว่
NVT: SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	Medium	ยังไม่ได้แก้ไข
NVT: HTTP Debugging Methods (TRACE/TRACK) Enabled	Medium	ไม่พบช่องโหว่

ผลการเปรียบเทียบช่องโหว่ที่พบโดยเครื่องมือ OpenVAS ก่อนการแก้ไข และหลังการแก้ไข พบว่ามีช่องโหว่เพิ่มขึ้น โดยเฉพาะช่องโหว่ระดับ High ดังตารางที่ 4-16

ตารางที่ 4-16 รายการเปรียบเทียบช่องโหว่ของ OpenVAS ก่อนการแก้ไข และหลังการแก้ไข

OpenVAS	จำนวนช่องโหว่ที่พบทั้งหมด	Critical	High	Medium	Low
BEFORE	72	-	33	39	-
AFTER	46	-	13	32	1

ตารางที่ 4-17 รายการช่องโหว่ใหม่ที่ตรวจพบหลังการแก้ไข ของเครื่องมือ OpenVAS

ช่องโหว่ที่ตรวจพบใหม่	ความรุนแรง
NVT: Webmin < 1.997 XSS Vulnerability	สูง (High)
NVT: Webmin 1.880 Information Disclosure Vulnerability	สูง (High)
NVT: Webmin <= 1.994 Multiple Vulnerabilities	สูง (High)
NVT: Webmin <= 1.991 Privilege Escalation Vulnerability	สูง (High)
NVT: Webmin <= 1.983 RCE Vulnerability	สูง (High)
NVT: Webmin <= 1.941 RCE Vulnerability	สูง (High)
NVT: Webmin <= 1.930 XXE Vulnerability	สูง (High)
NVT: Webmin < 1.930 RCE Vulnerability	สูง (High)
NVT: Webmin <= 1.984 Multiple Vulnerabilities	สูง (High)
NVT: Webmin <= 1.941 RCE Vulnerability	สูง (High)
NVT: SSL/TLS: Report Weak Cipher Suites	ปานกลาง (Medium)
NVT: SSL/TLS: Report Weak Cipher Suites	ปานกลาง (Medium)
SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection	ปานกลาง (Medium)
NVT: Joomla! XSS Vulnerability (20250102)	ปานกลาง (Medium)
NVT: Joomla! Access Control Vulnerability (20250103)	ปานกลาง (Medium)
NVT: Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)	ปานกลาง (Medium)
NVT: Weak Encryption Algorithm(s) Supported (SSH)	ปานกลาง (Medium)
NVT: Joomla! XSS Vulnerability (20250102)	ปานกลาง (Medium)
NVT: Joomla! Access Control Vulnerability (20250103)	ปานกลาง (Medium)
NVT: Cleartext Transmission of Sensitive Information via HTTP	ปานกลาง (Medium)
NVT: Webmin < 2.003 XSS Vulnerability	ปานกลาง (Medium)
NVT: Webmin <= 1.941 Multiple XSS Vulnerabilities	ปานกลาง (Medium)
NVT: Webmin <= 1.995 XSS Vulnerability	ปานกลาง (Medium)
NVT: SSL/TLS: Report Weak Cipher Suites	ปานกลาง (Medium)
NVT: SSL/TLS: Certificate Expired	ปานกลาง (Medium)
NVT: Weak MAC Algorithm(s) Supported (SSH)	ต่ำ (Low)

4.5.2 ผลการสแกนด้วย Nessus ครั้งที่ 2

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ Nessus ในการทดสอบการสแกนครั้งที่ 2 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 19 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีความรุนแรงระดับวิกฤต 4 ช่องโหว่ ช่องโหว่ที่มีระดับความรุนแรงสูง 4 ช่องโหว่ ช่องโหว่ที่มีระดับความรุนแรงปานกลาง 9 ช่องโหว่และ ช่องโหว่ที่มีระดับความรุนแรงต่ำ 2 ช่องโหว่ แสดงดังภาพที่ 4-23

ภาพที่ 4-23 ผลการสแกนของโปรแกรม Nessus (รอบที่ 2)

โดยระดับความรุนแรงของ CVSS v3.0 (Common Vulnerability Scoring System) ใช้เกณฑ์การให้คะแนนได้ 4 ระดับคือ

Critical (9.0-10) – มีผลกระทบร้ายแรง ควรแก้ไขทันที

High (7.0 – 8.9) – มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย

Medium (4.0 - 6.9) – มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี

Low (0.0-3.9) – มีผลกระทบต่ำ มีความมั่นคงปลอดภัย ไม่สามารถโจมตีได้ง่าย

ตารางที่ 4-18 รายการแก้ไขช่องโหว่ที่ตรวจพบจากเครื่องมือ Nessus (รอบที่2)

ช่องโหว่	ความรุนแรง	การแก้ไข
SSL Medium Strength Cipher Suites Supported (SWEET32)	High	ไม่พบช่องโหว่
HTTP TRACE / TRACK Methods Allowed	Medium	ยังไม่ได้แก้ไข
HTTP TRACE / TRACK Methods Allowed	Medium	ยังไม่ได้แก้ไข
JQuery 1.2 < 3.5.0 Multiple XSS	Medium	ยังไม่ได้แก้ไข
SSL Certificate Cannot Be Trusted	Medium	ยังไม่ได้แก้ไข
TLS Version 1.0 Protocol Detection	Medium	ไม่พบช่องโหว่
TLS Version 1.1 Deprecated Protocol	Medium	ไม่พบช่องโหว่

ผลการเปรียบเทียบช่องโหว่ที่พบโดยเครื่องมือ Nessus ก่อนการแก้ไข และหลังการแก้ไข พบว่ามีช่องโหว่เพิ่มขึ้น โดยเฉพาะช่องโหว่ระดับ Critical และ High ดังตารางที่ 4-19

ตารางที่ 4-19 รายการเปรียบเทียบช่องโหว่ของ Nessus ก่อนการแก้ไข และหลังการแก้ไข

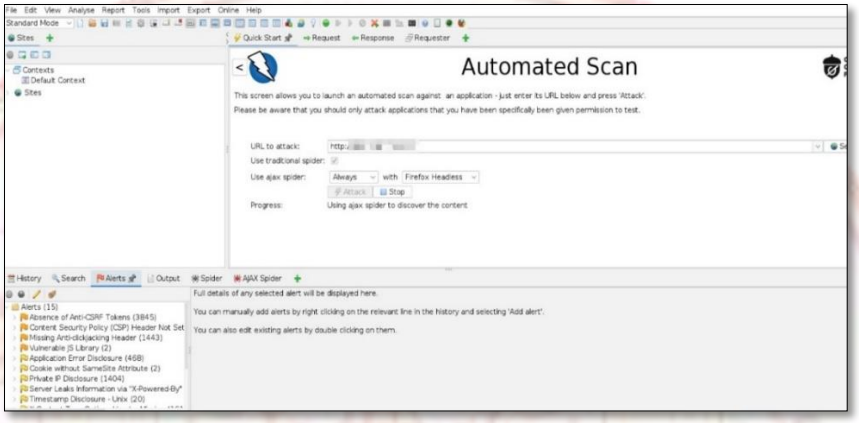
Nessus	จำนวนช่องโหว่ที่พบทั้งหมด	Critical	High	Medium	Low
BEFORE	7	-	1	6	-
AFTER	18	4	4	8	2

ตารางที่ 4-20 รายการช่องโหว่ใหม่ที่ตรวจพบหลังการแก้ไข ของเครื่องมือ Nessus

ช่องโหว่ที่ตรวจพบใหม่	**ความรุนแรง**
PHP 5.6.x < 5.6.34 Stack Buffer Overflow	วิกฤต (Critical)
PHP 5.6.x < 5.6.40 Multiple vulnerabilities.	วิกฤต (Critical)
PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.	วิกฤต (Critical)
PHP Unsupported Version Detection	วิกฤต (Critical)
PHP 5.6.x < 5.6.36 Multiple Vulnerabilities	สูง (High)
PHP 5.6.x < 5.6.37 exif_thumbnail_extract (DoS)	สูง (High)
PHP 5.6.x < 5.6.39 Multiple vulnerabilities	สูง (High)
PHP < 7.3.24 Multiple Vulnerabilities	สูง (High)
PHP 5.6.x < 5.6.33 Multiple Vulnerabilities	ปานกลาง (Medium)
PHP 5.6.x < 5.6.38 Transfer-Encoding Parameter XSS	ปานกลาง (Medium)
SSH Terrapin Prefix Truncation Weakness (CVE-2023-48795)	ปานกลาง (Medium)
PHP < 7.3.28 Email Header Injection	ปานกลาง (Medium)
PHP 5.6.x < 5.6.35 Security Bypass Vulnerability	ปานกลาง (Medium)
SSH Server CBC Mode Ciphers Enabled	ต่ำ (Low)
SSH Weak Key Exchange Algorithms Enabled	ต่ำ (Low)

4.5.3 ผลการสแกนด้วย OWASP ZAP ครั้งที่ 2

จากการทดสอบเว็บแอปพลิเคชันขององค์กรโดยใช้เครื่องมือ OWASP ZAP ในการทดสอบการสแกนครั้งที่ 2 พบว่าเว็บแอปพลิเคชันขององค์กรมีช่องโหว่ทั้งหมด 10 ช่องโหว่ ประกอบด้วยช่องโหว่ที่มีระดับความรุนแรงปานกลาง 4 ช่องโหว่ และช่องโหว่ที่มีระดับความรุนแรงต่ำ 6 ช่องโหว่ แสดงดังภาพที่ 4-24



ภาพที่ 4-24 ผลการสแกนของโปรแกรม OWASP ZAP (รอบที่ 2)

ตารางที่ 4-21 รายการแก้ไขช่องโหว่ที่ตรวจพบจาก OWASP ZAP (รอบที่2)

ช่องโหว่	ความรุนแรง	การแก้ไข
Absence of Anti-CSRF Tokens	Medium	ยังไม่ได้แก้ไข
Content Security Policy (CSP) Header Not Set	Medium	ยังไม่ได้แก้ไข
Missing Anti-clickjacking Header	Medium	ยังไม่ได้แก้ไข
Vulnerable JS Library	Medium	ยังไม่ได้แก้ไข
Application Error Disclosure	Low	ยังไม่ได้แก้ไข
Cookie Without Secure Flag	Low	ไม่พบช่องโหว่
Cookie without SameSite Attribute	Low	ยังไม่ได้แก้ไข
Private IP Disclosure	Low	ยังไม่ได้แก้ไข
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	ยังไม่ได้แก้ไข
Strict-Transport-Security Header Not Set	Low	ไม่พบช่องโหว่
Timestamp Disclosure - Unix	Low	ยังไม่ได้แก้ไข
X-Content-Type-Options Header Missing	Low	ยังไม่ได้แก้ไข

ผลการเปรียบเทียบช่องโหว่ที่พบโดยเครื่องมือ OWASP ZAP ก่อนการแก้ไข และหลังการแก้ไข ไม่พบช่องโหว่ระดับ Critical และ High เพิ่มขึ้น ดังตารางที่ 4-22

ตารางที่ 4-22 รายการเปรียบเทียบช่องโหว่ของ OWASP ZAP ก่อนการแก้ไข และหลังการแก้ไข

OWASP ZAP	จำนวนช่องโหว่ที่พบทั้งหมด	Critical	High	Medium	Low
BEFORE	12	-	-	4	8
AFTER	10	-	-	4	6

4.6 การเปรียบเทียบประสิทธิภาพของเครื่องมือสแกนช่องโหว่

การประเมินประสิทธิภาพเครื่องมือสแกนช่องโหว่เว็บแอปพลิเคชันขององค์กรทั้ง 3 เครื่องมือนั้น จำเป็นต้องทำการปรับผลการประเมินระดับความรุนแรงของช่องโหว่ โดยใช้เกณฑ์ให้คะแนนเดียวกัน ปัจจุบันเครื่องมือ OpenVAS และ Nessus จะใช้มาตรฐานเดียวกันในการประเมินความรุนแรงของช่องโหว่คือ CVSS (Common Vulnerability Scoring System) แต่ทั้งสองเครื่องมือใช้คนละเวอร์ชัน ซึ่งส่งผลต่อการจัดระดับความรุนแรงและคะแนนของช่องโหว่อย่างมีนัยสำคัญ โดย OpenVAS ใช้มาตรฐาน CVSS เวอร์ชัน 2.0 ซึ่งมีการจัดระดับความรุนแรงเป็น None, Low, Medium และ High โดยพิจารณาจากเมตริกพื้นฐาน เช่น ความยากในการเข้าถึง และผลกระทบต่อระบบ ส่วน Nessus ใช้ CVSS เวอร์ชัน 3.0 ซึ่งมีความละเอียดมากขึ้น ด้วยการเพิ่มเมตริก เช่น ระดับสิทธิ์ที่ต้องใช้ในการโจมตี (Privileges Required) และขอบเขตของผลกระทบ (Scope) พร้อมทั้งเพิ่มระดับความรุนแรง Critical สำหรับช่องโหว่ที่มีคะแนน 9.0–10.0 เพื่อสะท้อนความรุนแรงสูงสุด ความแตกต่างนี้ทำให้ช่องโหว่เดียวกันอาจได้รับการจัดลำดับไม่เท่ากันในสองเครื่องมือ เช่น ช่องโหว่ที่ OpenVAS ประเมินว่าอยู่ในระดับ High อาจถูก Nessus จัดอยู่ในระดับ Critical ซึ่งสะท้อนให้เห็นว่าแม้ใช้มาตรฐานเดียวกัน แต่เวอร์ชันของ CVSS ก็มีผลต่อการประเมินความเสี่ยงอย่างชัดเจน ส่วน OWASP ZAP ใช้ระบบจัดระดับความเสี่ยงแบบง่าย (High, Medium, Low, Informational) ซึ่งไม่สามารถแปลงเป็นคะแนน CVSS ได้โดยตรง แต่สามารถประมาณค่าเทียบเคียงกับคะนน CVSS ได้

ในการค้นคว้าอิสระนี้ ผู้วิจัยจะปรับเกณฑ์คะแนนของแต่ละเครื่องมือมาเป็นเกณฑ์ให้คะแนนอ้างอิงตามระดับความรุนแรงของ CVSS v3.0 (Common Vulnerability Scoring System) มี 4 ระดับ ดังนี้

Critical (9.0-10) – มีผลกระทบร้ายแรง ควรแก้ไขทันที

High (7.0 – 8.9) – มีผลกระทบรุนแรง สามารถถูกโจมตีได้ง่าย
Medium (4.0 - 6.9) – มีผลกระทบปานกลาง อาจต้องใช้ทักษะขั้นสูงในการโจมตี
Low (0.0-3.9) – มีผลกระทบต่ำ มีความมั่นคงปลอดภัย ไม่สามารถโจมตีได้ง่าย

ที่ตรวจ CVSS v3.0 นี้เป็นเกณฑ์ที่ Nessus ใช้วัดระดับความรุนแรงในปัจจุบัน ดังนั้น หากต้องปรับเกณฑ์ของ OpenVAS ที่ใช้คะแนนระดับความรุนแรงของ CVSS v2.0 และมีการระบุค่าคะแนนไว้ จะสามารถนำมาเปรียบเทียบกับเกณฑ์คะแนนระดับความรุนแรงของ CVSS v3.0 ได้โดยตรง ส่วนเครื่องมือ OWASP ZAP แม้จะไม่สามารถแปลงระดับความรุนแรงไปเป็นคะแนน CVSS v3.0 ได้โดยตรง เพราะ OWASP ZAP ใช้ระบบจัดระดับแบบง่าย ส่วน CVSS v3.0 ใช้โมเดลคำนวณบนหลายปัจจัย เช่น Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR) User Interaction (UI), Scope (S) และ Confidentiality (C), Integrity (I), Availability (A) เป็นต้น แต่สามารถใช้ในแนวทางการแปลงคะแนนเทียบเคียงจาก [19] ได้ เมื่อทำการปรับเกณฑ์ให้เหมือนกันแล้ว พบว่าเครื่องมือ OpenVAS มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 72 ช่องโหว่ แบ่งเป็นความรุนแรงระดับวิกฤต 12 ช่องโหว่, ความรุนแรงระดับสูง 18 ช่องโหว่, ความรุนแรงระดับปานกลาง 42 ช่องโหว่ หลังจากทำการแก้ไขแล้ว มีจำนวนช่องโหว่ลดลงเหลือ 46 ช่องโหว่ โดยช่องโหว่ที่มีความรุนแรงระดับวิกฤตลดลงจาก 12 ช่องโหว่ เหลือ 3 ช่องโหว่, ช่องโหว่ที่มีความรุนแรงระดับสูงลดลงจาก 18 ช่องโหว่ เหลือ 10 ช่องโหว่, ช่องโหว่ที่มีความรุนแรงระดับปานกลางลดลงจาก 42 ช่องโหว่ เหลือ 32 ช่องโหว่ และมีการตรวจพบช่องโหว่ที่มีความรุนแรงระดับต่ำ 1 ช่องโหว่ ซึ่งเครื่องมือ OpenVAS มีเกณฑ์การให้คะแนนระดับความรุนแรงของ CVSS v2.0

เครื่องมือ Nessus มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 7 ช่องโหว่ แบ่งเป็นความรุนแรงระดับสูง 1 ช่องโหว่, ความรุนแรงระดับปานกลาง 6 ช่องโหว่ หลังจากทำการแก้ไขแล้ว มีจำนวนช่องโหว่เพิ่มขึ้นเป็น 18 ช่องโหว่ โดยช่องโหว่ที่มีความรุนแรงระดับสูงมากเพิ่มขึ้น 4 ช่องโหว่, ช่องโหว่ที่มีความรุนแรงระดับสูงเพิ่มขึ้นจาก 1 ช่องโหว่เป็น 4 ช่องโหว่ ช่องโหว่ที่มีความรุนแรงระดับปานกลางเพิ่มขึ้นจาก 6 ช่องโหว่เป็น 8 ช่องโหว่ และมีการตรวจพบช่องโหว่ที่มีความรุนแรงระดับต่ำ 1 ช่องโหว่ ช่องโหว่ที่เพิ่มขึ้นหลังจากการแก้ไข อาจเกิดจากการอัปเดตแพตช์หรือมีการอัปเดตช่องโหว่อยู่ตลอดของเครื่องมือที่ตรวจพบ หรืออาจเป็นการตรวจสอบด้วยเครื่องมือที่ละเอียดมากขึ้น ซึ่งเครื่องมือ Nessus มีเกณฑ์การให้คะแนนระดับความรุนแรงของ CVSS v3.0

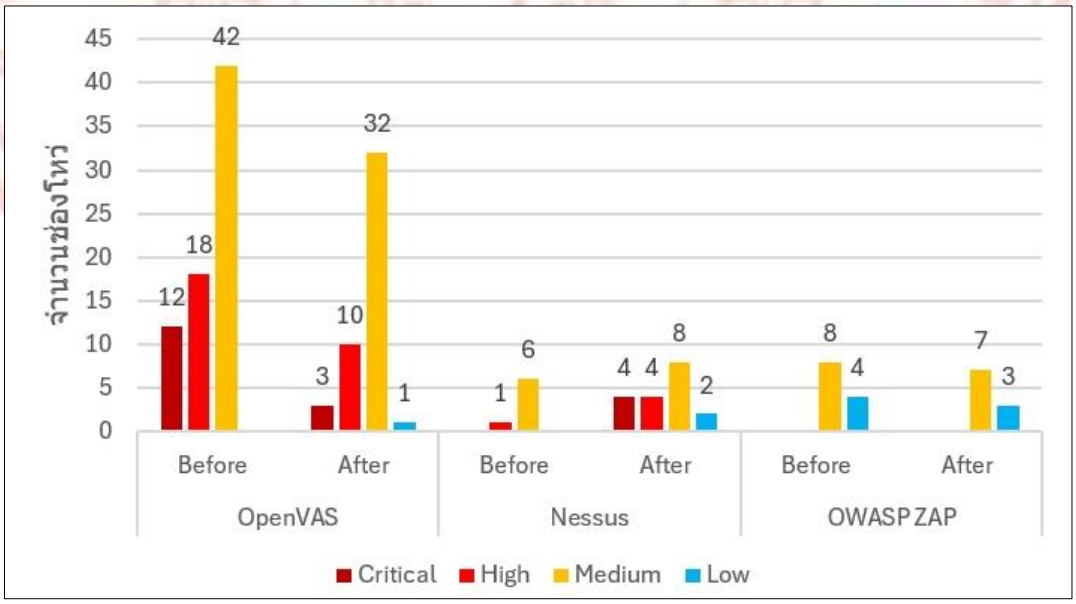
OWASP ZAP มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 12 ช่องโหว่ แบ่งเป็นช่องโหว่ที่มี

ความรุนแรงระดับปานกลาง 8 ช่องโหว่ ช่องโหว่ที่มีความรุนแรงระดับต่ำ 4 ช่องโหว่ หลังจากทำการแก้ไขแล้ว มีจำนวนช่องโหว่ลดลงเหลือ 10 ช่องโหว่ โดยช่องโหว่ที่มีความรุนแรงระดับปานกลางลดลงจาก 8 ช่องโหว่ เหลือ 7 ช่องโหว่ และช่องโหว่ที่มีความรุนแรงระดับต่ำลดลงจาก 4 ช่องโหว่ เหลือ 3 ช่องโหว่

ผลการเปรียบเทียบช่องโหว่ในแต่ละระดับความรุนแรงโดยใช้เกณฑ์วัด CVSS 3.0 เช่นเดียวกัน แสดงดังตารางที่ 4-23

ตารางที่ 4-23 สรุปจำนวนช่องโหว่ที่ตรวจพบแยกตามเครื่องมือและระดับความรุนแรง (CVSS v3.0)

Tools	การทดสอบ	จำนวนช่องโหว่ที่พบทั้งหมด	Critical	High	Medium	Low
OpenVAS	BEFORE	72	12	18	42	-
	AFTER	46	3	10	32	1
Nessus	BEFORE	7	-	1	6	-
	AFTER	18	4	4	8	2
OWASP ZAP	BEFORE	12	-	-	8	4
	AFTER	10	-	-	7	3

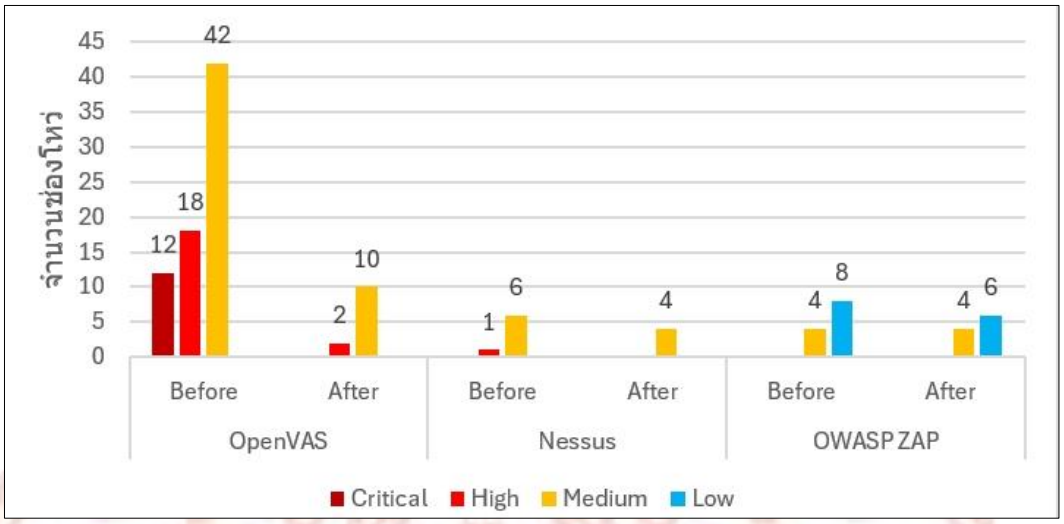


ภาพที่ 4-25 ผลการเปรียบเทียบช่องโหว่ที่ตรวจพบแต่ละเครื่องมือก่อนและหลังการแก้ไขช่องโหว่ (ตามเกณฑ์ CVSS v3.0)

หากพิจารณาความสามารถในการแก้ไขช่องโหว่ที่พบในการสแกนครั้งแรก และไม่นับรวมช่องโหว่ที่เกิดขึ้นใหม่หลังจากการแก้ไข สามารถสรุปได้ว่าเมื่อมีการแก้ไขช่องโหว่แล้วพบว่า เครื่องมือ OpenVAS มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 72 ช่องโหว่ หลังการแก้ไขพบช่องโหว่ที่ตรวจพบจากการสแกนครั้งที่ 1 12 ช่องโหว่ ช่องโหว่ถูกแก้ไขไป 60 ช่องโหว่ เครื่องมือ Nessus มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 7 ช่องโหว่ หลังการแก้ไข พบช่องโหว่ที่ตรวจพบจากการสแกนครั้งที่ 1 3 ช่องโหว่ ช่องโหว่ถูกแก้ไขไป 4 ช่องโหว่ และ เครื่องมือ OWASP ZAP มีการตรวจพบช่องโหว่ก่อนการแก้ไขทั้งหมด 12 ช่องโหว่ หลังการแก้ไขพบช่องโหว่ที่ตรวจพบจากการสแกนครั้งที่ 1 10 ช่องโหว่ ช่องโหว่ถูกแก้ไขไป 2 ช่องโหว่ แสดงดังตารางที่ 4-24

ตารางที่ 4-24 สรุปจำนวนช่องโหว่ที่ตรวจก่อนและหลังการแก้ไขช่องโหว่จากทั้ง 3 เครื่องมือ (ไม่รวมช่องโหว่ที่เกิดขึ้นใหม่)

Tools	การทดสอบ	จำนวนช่องโหว่	Critical	High	Medium	Low
OpenVAS	BEFORE	72	12	18	42	-
	AFTER	12	-	2	10	-
	ถูกแก้ไข	60	12	16	32	-
Nessus	BEFORE	7	-	1	6	-
	AFTER	4	-	-	4	-
	ถูกแก้ไข	3	-	1	2	
OWASP ZAP	BEFORE	12	-	-	8	4
	AFTER	10	-	-	7	3
	ถูกแก้ไข	2	-	-	1	1



ภาพที่ 4-26 ผลการเปรียบเทียบช่องโหว่ที่ตรวจพบแต่ละเครื่องมือก่อนและหลังการแก้ไข (ไม่รวมช่องโหว่ที่เกิดขึ้นใหม่)

นอกเหนือจากมิติความสามารถในการตรวจจับช่องโหว่แล้ว ต้องพิจารณาเปรียบเทียบในมิติอื่น เช่น ระยะเวลาในการสแกน และความแม่นยำของผลลัพธ์ ดังแสดงในตารางที่ 4-25

ตารางที่ 4-25 รายการเปรียบเทียบผลการสแกนช่องโหว่

คุณสมบัติ	OpenVAS	Nessus	OWASP ZAP
จำนวนช่องโหว่ที่พบก่อนการแก้ไข	72	7	12
จำนวนช่องโหว่ที่พบหลังการแก้ไข	46	18	10
จำนวนช่องโหว่ที่ตรวจพบใหม่	26	15	-
จำนวนการเปลี่ยนแปลงของโหว่	ลดลง 26	เพิ่มขึ้น 9	ลดลง 2
ช่องโหว่ที่ยังไม่ได้รับการแก้ไข	12	4	10
เวลาที่ใช้ในการสแกน (โดยเฉลี่ย)	40 นาที	25 นาที	30 นาที
การใช้งาน	ปานกลาง	ง่าย	ง่าย
การตรวจจับ Security Misconfiguration	ดีมาก	ปานกลาง	ปานกลาง
การตรวจจับ Injection	ดี	ดีมาก	ปานกลาง
ความแม่นยำ	ปานกลาง	มากที่สุด	ปานกลาง
เกณฑ์การให้คะแนน	CVSS v2.0	CVSS v3.0	Risk Rating

จากการศึกษาประสิทธิภาพของเครื่องมือสแกนช่องโหว่ OpenVAS, Nessus และ OWASP ZAP แต่ละเครื่องมือจุดแข็งและจุดอ่อนที่แตกต่างกัน โดยพิจารณาจาก ความสามารถในการตรวจจับช่องโหว่, ความแม่นยำ, ระยะเวลาในการสแกน และการใช้งาน สามารถสรุปได้ดังนี้

OpenVAS เหมาะสำหรับการตรวจจับการตั้งค่าความมั่นคงปลอดภัยผิดพลาด (Security Misconfiguration) สามารถวิเคราะห์การตั้งค่าที่ไม่ปลอดภัยของเซิร์ฟเวอร์และระบบเครือข่ายได้ดี สามารถตรวจจับช่องโหว่ได้หลากหลาย ใช้เวลาในการสแกนนานที่สุด และใช้งานยาก จำเป็นต้องมีความรู้ทางเทคนิคในการใช้เครื่องมือ ส่วน Nessus มีความแม่นยำสูงสุด ใช้เวลาสแกนน้อยที่สุดมีความรวดเร็วและมีประสิทธิภาพ ใช้งานง่าย และเหมาะที่จะใช้กับองค์กรพึ่งจะเริ่มต้นการตรวจจับ Misconfiguration ยังไม่ละเอียด OWASP ZAP เป็นเครื่องมือที่เหมาะสำหรับการทดสอบ Web Application Security แต่ในการตรวจจับ Misconfiguration ยังไม่ละเอียด ความแม่นยำในการตรวจจับยังอยู่ในระดับปานกลาง จุดแข็งและจุดอ่อนของแต่ละเครื่องมือ ช่วยให้สามารถเลือกใช้งานเครื่องมือที่เหมาะสมกับความต้องการขององค์กรได้อย่างมีประสิทธิภาพ แสดงดังตารางที่ 4-26

ตารางที่ 4-26 รายการเปรียบเทียบจุดแข็งและจุดอ่อนของแต่ละเครื่องมือ

เครื่องมือ	จุดแข็ง	จุดอ่อน
OpenVAS	ตรวจจับช่องโหว่ได้มากที่สุด เหมาะกับการตรวจจับ Security Misconfiguration มีฐานข้อมูลช่องโหว่ขนาดใหญ่ NVT และ CVE และมีความละเอียด	ใช้เวลาสแกนนานที่สุด (40 นาที) ใช้งานยาก ต้องใช้เทคนิคขั้นสูง
Nessus	มีความแม่นยำสูงสุด ใช้เวลาสแกนน้อยที่สุด (25 นาที) ใช้งานง่าย	ตรวจจับ Security Misconfiguration ได้ปานกลาง จำนวนช่องโหว่ที่พบเพิ่มขึ้นหลังการแก้ไข
OWASP ZAP	เหมาะสำหรับการทดสอบ Web Application Security ใช้งานง่าย	ตรวจจับช่องโหว่ได้น้อย ความแม่นยำปานกลาง

4.7 การพัฒนากลยุทธ์ป้องกันการโจมตีองค์กร

ในปัจจุบัน การโจมตีทางไซเบอร์ผ่านเว็บแอปพลิเคชันเป็นภัยคุกคามที่สำคัญต่อองค์กร ช่องโหว่ที่ตรวจพบในระบบอาจจะเป็นช่องทางให้ผู้ไม่หวังดีเข้าถึงข้อมูลที่สำคัญเพื่อทำลายระบบ ดังนั้นองค์กรควรมีแนวทางการป้องกันภัยคุกคามทางไซเบอร์จากเว็บแอปพลิเคชัน รวมถึงการพัฒนากลยุทธ์ป้องกันการโจมตีที่ครอบคลุมทั้ง โดยแบ่งเป็นหมวดหมู่ ดังนี้

4.7.1 การป้องกัน (Prevention)

การตั้งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์และเครือข่ายการตั้งค่าที่ไม่ปลอดภัย ซึ่งอาจจะเป็นช่องทางให้ผู้ไม่หวังดีเข้าถึงข้อมูลสำคัญ วิธีป้องกันคือใช้ SSL/TLS Encryption สำหรับการสื่อสาร ตรวจสอบและใช้ Security Headers เช่น HSTS (HTTP Strict Transport Security) เปิดใช้งาน WAF (Web Application Firewall) และทำการตรวจสอบการตั้งค่าความมั่นคงปลอดภัยของเซิร์ฟเวอร์อย่างสม่ำเสมอเพื่อป้องกันความเสี่ยงที่อาจเกิดขึ้น และการอัปเดตหรือทำการแพตช์ระบบสม่ำเสมอ เป็นกระบวนการสำคัญในการป้องกันช่องโหว่ด้านความมั่นคงปลอดภัยที่อาจถูกใช้ในการโจมตีทางไซเบอร์ จัดลำดับความเสี่ยงของแพตช์ตามระดับความรุนแรง โดยผู้พัฒนาซอฟต์แวร์จะออก Security Patch หรืออัปเดตเพื่อแก้ไขปัญหาด้านความมั่นคงปลอดภัยที่พบ การไม่อัปเดตระบบเป็นประจำอาจทำให้ระบบมีความเสี่ยงต่อการถูกโจมตี เช่น Zero-Day Exploit, Ransomware หรือ Privilege Escalation การอัปเดตและแพตช์ระบบอย่างสม่ำเสมอช่วยลดความเสี่ยงจากช่องโหว่ด้านความมั่นคงปลอดภัย และช่วยให้องค์กรมีความมั่นคงปลอดภัยมากขึ้น ควรมีกระบวนการจัดการการอัปเดตระบบที่ดี รวมถึงการทดสอบแพตช์ก่อนติดตั้ง เพื่อป้องกันผลกระทบที่อาจเกิดขึ้นกับระบบหลัก

4.7.2 การตรวจจับ (Detection) มีระบบเฝ้าระวังความมั่นคงปลอดภัย (Security Monitoring) โดยการใช้ SIEM (Security Information and Event Management) เพื่อตรวจสอบ Log เช่น ตรวจสอบพฤติกรรมผิดปกติ เช่น การ Login เข้าระบบที่ไม่สำเร็จหลายครั้ง มีการทำการสแกนระบบอย่างสม่ำเสมอเพื่อตรวจสอบช่องโหว่ที่อาจเกิดขึ้นใหม่ได้

4.7.3 การตอบสนองและการฟื้นฟู (Response and Recovery) จัดทำแผนรับมือกับเหตุการณ์ด้านความมั่นคงปลอดภัยที่จะเกิดขึ้น เช่น ระบุขั้นตอนการรับมือหากถูกโจมตี และกำหนดทีม Incident Response และ Contact List มีการสำรองข้อมูลอัตโนมัติ และทดสอบการกู้คืนระบบ

มีการฝึกอบรมหรือให้ความรู้บุคลากรในองค์กรเกี่ยวกับ Social Engineering, Phishing ภัยคุกคามต่างๆ และ การใช้เครื่องมือตรวจสอบช่องโหว่อยู่เป็นประจำ

บทที่ 5
อภิปรายผลและสรุปผล

จากการดำเนินงานการการประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพสามารถสรุปอภิปรายผลการวิจัย แบ่งได้เป็น 3 ส่วน ดังนี้

5.1 อภิปรายผลการวิจัย

5.2 สรุปผลการวิจัย

5.3 ข้อเสนอแนะ

5.1 อภิปรายผลการวิจัย

จากการวิเคราะห์ผลการศึกษาการประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพในองค์กรแห่งหนึ่ง จากผลการวิเคราะห์การสแกนช่องโหว่ของเว็บแอปพลิเคชันองค์กรทั้ง 2 รอบ พบว่า เครื่องมือ OpenVAS มีช่องโหว่เก่าที่พบในการสแกนรอบแรกลดลง 60 ช่องโหว่ และพบช่องโหว่ใหม่เพิ่มขึ้น 25 ช่องโหว่ เครื่องมือ Nessus มีช่องโหว่เก่าที่พบในการสแกนรอบแรกลดลง 3 ช่องโหว่ และพบช่องโหว่ใหม่เพิ่มขึ้น 12 ช่องโหว่ และเครื่องมือ OWAPS ZAP มีช่องโหว่เก่าที่พบในการสแกนรอบแรกลดลง 2 ช่องโหว่ และไม่พบช่องโหว่ใหม่เพิ่มขึ้น นั่นแสดงว่าหลังจากที่ได้ทำการแก้ไขช่องโหว่แล้วช่องโหว่ที่มีระดับความรุนแรงหายไปบางส่วนทำให้ลดระดับความเสี่ยงที่จะถูกโจมตีลดลงทำให้เว็บแอปพลิเคชันมีความมั่นคงปลอดภัยเพิ่มมากขึ้น และตรวจพบช่องโหว่ที่เกิดขึ้นใหม่ ซึ่งเกิดจากการอัปเดตช่องโหว่ที่เพิ่งถูกเปิดเผย ผลกระทบจากการอัปเดตซอฟต์แวร์ เพื่อป้องกันปัญหานี้องค์กรควร ทดสอบระบบหรือทำการสแกนช่องโหว่อย่างต่อเนื่อง และการที่ทั้ง 3 เครื่องมือสามารถตรวจจับช่องโหว่ได้ไม่เท่ากันนั้นอาจเกิดจากหลายปัจจัย โดยปัจจัยที่สำคัญคือ หลักการทำงานของแต่ละเครื่องมือ หากฐานข้อมูลมีความหลากหลายและมีการอัปเดตอย่างสม่ำเสมอ เช่น Nessus หรือ OpenVAS ก็มีสามารถที่จะตรวจจับช่องโหว่ได้มากกว่าเครื่องมือที่ฐานข้อมูลไม่ครอบคลุม ส่วนเครื่องมือ OWASP ZAP จะเน้นการวิเคราะห์เพื่อทดสอบการทำงานของเว็บแอปพลิเคชัน ซึ่งจะมีประสิทธิภาพในการตรวจสอบเฉพาะด้านแต่ก็อาจตรวจไม่พบช่องโหว่ในระดับเครือข่ายหรือระบบปฏิบัติการได้ นอกจากนี้ ความสามารถในการตั้งค่าการสแกน ก็เป็นปัจจัยที่ทำให้เครื่องมือ OpenVAS ตรวจจับได้มากกว่า Nessus และ OWASP ZAP

OpenVAS สามารถตรวจจับช่องโหว่ได้ดี แต่มีข้อจำกัดในการตั้งค่าผ่าน Task Wizard ทำให้ไม่สามารถดึงศักยภาพของฐานข้อมูล signature ออกมาได้เต็มที่ ถึงแม้เครื่องมือจะมีฐานข้อมูล

ครอบคลุม แต่หากไม่ตั้งค่าอย่างเหมาะสม ประสิทธิภาพก็ลดลงได้ ซึ่งควรต้องมีผู้เชี่ยวชาญในการใช้เครื่องมือเพื่อให้สามารถใช้เครื่องมือได้อย่างมีประสิทธิภาพ ซึ่งสอดคล้องกับข้อเสนอแนะใน [10] ส่วน OWASP ZAP ใช้แนวทางแบบ active scanning และ fuzzing ซึ่งไม่พึ่งพา signature เป็นหลัก แต่จะส่ง request ที่มี payload หลายรูปแบบไปยังเว็บแอปพลิเคชันเพื่อตรวจสอบปฏิกิริยา เช่น XSS หรือ SQLi ซึ่งทำให้ ZAP เหมาะกับการหาช่องโหว่ในแอปพลิเคชันเลเยอร์มากกว่าเน็ตเวิร์กเลเยอร์ ทำให้ OWASP ZAP ตรวจพบช่องโหว่น้อยกว่า OpenVAS หรือ Nessus

อย่างไรก็ตาม ผลการเปรียบเทียบประสิทธิภาพของเครื่องมือทั้ง 3 ตัวนี้ เกิดจากการทดสอบเครื่องมือกับระบบสารสนเทศในบริษัทที่ใช้ในการศึกษานี้เท่านั้น หากสภาพแวดล้อม หรือระบบที่ใช้ทดสอบ มีการตั้งค่าระบบปฏิบัติการ ซอฟต์แวร์ หรือบริการต่างกัน ก็อาจให้ข้อสรุปที่ต่างอ ออกไป นอกจากนี้หากองค์กรมีการติดตั้ง Firewall, Web Application Firewall หรือระบบ IDS/IPS ก็อาจจำกัดการสแกนหรือจำกัดข้อมูลที่เครื่องมือเข้าถึงได้ ทำให้ผลลัพธ์ของการตรวจจับไม่ได้สะท้อนศักยภาพจริงของเครื่องมือที่ควรจะเป็น หรือในบางกรณี เครื่องมืออาจไม่สามารถเข้าถึง endpoint บางประเภทได้เลยหากไม่ได้รับสิทธิ์แบบ authenticated ดังนั้น ผลการทดสอบจึงไม่ได้ขึ้นอยู่กับเครื่องมืออย่างเดียว แต่ขึ้นอยู่กับบริบทที่เครื่องมือถูกใช้งานด้วย

5.2 สรุปผลการวิจัย

ในการประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชัน ผู้วิจัยได้ทำการทดสอบโดยการคัดเลือกเว็บแอปพลิเคชันของบริษัทเอกชนแห่งหนึ่งมาใช้เป็นกรณีศึกษา โดยได้ดำเนินการสแกนช่องโหว่โดยใช้เครื่องมือแบบโอเพ่นซอร์ส จำนวน 3 เครื่องมือ คือ OpenVAS, Nessus และ OWASP ZAP ภายหลังจากการสแกนครั้งที่ 1 ผู้วิจัยได้ทำการเปรียบเทียบช่องโหว่ที่พบแต่ละเครื่องมือและทำการวิเคราะห์ความเสี่ยงที่จะเกิดขึ้นโดยจัดกลุ่มรายการความเสี่ยงด้านความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน ตาม OWASP Top Ten 2021 โดยพบว่าช่องโหว่ส่วนใหญ่จะเป็นประเภท Security Misconfiguration ที่สะท้อนถึงปัญหาในการกำหนดสิทธิ์หรือการตั้งค่าความมั่นคงปลอดภัยของระบบที่ไม่ถูกต้องอย่างชัดเจน ซึ่งจะต้องจัดทำรายงานการค้นพบช่องโหว่นำเสนอต่อเจ้าหน้าที่ที่ดูแลเว็บแอปพลิเคชันขององค์กร พร้อมทั้งนำเสนอแนวทางการแก้ไขช่องโหว่ที่เกิดขึ้นเพื่อให้สามารถแก้ไขปรับปรุงช่องโหว่ที่พบที่มีความเสี่ยงสูง

เมื่อเปรียบเทียบผลการสแกนครั้งที่ 1 และครั้งที่ 2 ภายหลังจากการแก้ไขช่องโหว่แล้ว พบว่าช่องโหว่ที่มีความรุนแรงสูงที่ได้รับการแก้ไขตามแนวทางที่นำเสนอไม่ถูกตรวจพบในการสแกนรอบที่ 2 แต่มีการตรวจพบช่องโหว่ใหม่ที่เกิดขึ้นมาซึ่งอาจเกิดจากการโจมตีที่เกิดขึ้นใหม่ หรือเวอร์ชันของซอฟต์แวร์ที่อัปเดตเพื่อแก้ไขช่องโหว่เดิมมีช่องโหว่ใหม่เพิ่มขึ้นมา ซึ่งเมื่อเปรียบเทียบทั้ง 3 เครื่องมือ พบว่า หากต้องเลือกเครื่องมือสแกนช่องโหว่สำหรับองค์กร เครื่องมือที่เหมาะสมและคุ้มค่าที่สุดคือ

OpenVAS เนื่องจาก เป็นเครื่องมือแบบ Open Source ที่ไม่มีค่าใช้จ่าย ถึงแม้จะต้องใช้ทักษะและเทคนิคขั้นสูงในการตั้งค่า แต่มีความสามารถในการตรวจจับช่องโหว่ได้อย่างครอบคลุม ทั้งในระดับระบบปฏิบัติการ เครือข่าย และแอปพลิเคชัน โดยอาศัยเทคนิคแบบ signature-based ที่เชื่อมโยงกับฐานข้อมูลช่องโหว่สากลอย่าง CVE ที่มีการอัปเดตอย่างต่อเนื่อง ซึ่งช่วยให้องค์กรสามารถระบุและประเมินความรุนแรงของช่องโหว่ได้อย่างมีประสิทธิภาพ นอกจากนี้ OpenVAS ยังสามารถปรับแต่งรูปแบบการสแกนให้เหมาะสมกับองค์กร จึงเหมาะสำหรับการใช้งานในภาพรวม แต่หากองค์กรมีการพัฒนาเว็บแอปพลิเคชันจำนวนมาก ควรเพิ่ม OWASP ZAP เพื่อเน้นการตรวจสอบช่องโหว่ในระดับแอปพลิเคชัน เช่น XSS หรือ SQL Injection โดยตรง และหากมีงบประมาณเพียงพอ องค์กรสามารถใช้ Nessus ควบคู่กันไปเพื่อความแม่นยำที่สูงขึ้นและความสะดวกรวดเร็วในการใช้งาน จากเหตุผลทั้งหมดนี้ จึงสรุปได้ว่า OpenVAS เป็นทางเลือกหลักที่เหมาะสมสำหรับองค์กรที่ต้องการความครอบคลุม ความคุ้มค่า และความสามารถในการปรับแต่งการใช้งานเพื่อรองรับการรักษาความมั่นคงปลอดภัย

นอกเหนือจากเครื่องมือแล้ว กลยุทธ์ที่สำคัญเพื่อป้องกันการโดนโจมตีหรือลดผลกระทบที่อาจจะเกิดขึ้นจากการใช้ประโยชน์จากช่องโหว่ องค์กรควรกำหนดนโยบายการทดสอบและการสแกนช่องโหว่อย่างสม่ำเสมอและต่อเนื่อง (ทุก 3 หรือ 6 เดือน) อัปเดตฐานข้อมูลช่องโหว่อย่างสม่ำเสมอ เพื่อให้สามารถปรับตัวให้ทันต่อภัยคุกคามใหม่ที่จะเกิดขึ้น ปรับปรุงกระบวนการแก้ไขช่องโหว่ ให้มีวิธีการที่สามารถแก้ไขช่องโหว่ได้อย่างรวดเร็วและมีประสิทธิภาพ เพื่อลดความเสี่ยงในการถูกโจมตี นอกจากนี้ องค์กรควรมีกลยุทธ์ในการสร้างความตะหนักรู้และมีการจัดอบรมให้กับบุคลากรเกี่ยวกับภัยคุกคามใหม่ ๆ และมีแนวทางปฏิบัติที่ดีในการป้องกันอย่างต่อเนื่อง โดยต้องมีแผนรับมือและตอบสนองเหตุการณ์ที่ชัดเจนและสามารถปฏิบัติได้จริง เมื่อพบช่องโหว่หรือเกิดเหตุการณ์โจมตี

5.3 ข้อเสนอแนะ

การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ มีข้อเสนอแนะ ดังนี้

1. ควรทำการสแกนช่องโหว่ภายในองค์กรทุกๆ 3 เดือนเพื่อตรวจสอบช่องโหว่ที่จะเกิดขึ้นใหม่ และหาแนวทางการป้องกันแก้ไข

2. มีการเพิ่มระบบ Real-Time Monitoring และ Incident Response เพื่อช่วยให้องค์กรสามารถตรวจจับและตอบสนองต่อภัยคุกคามทางไซเบอร์ได้อย่างรวดเร็วและมีประสิทธิภาพ

3. ควรมีการใช้ Penetration Testing Tools ควบคู่กับ Vulnerability Scanners เพื่อทดสอบและยืนยันว่าช่องโหว่ที่ตรวจพบนั้นสามารถถูกโจมตีได้จริงหรือไม่

มาตรการเหล่านี้จะช่วยให้องค์กรสามารถลดความเสี่ยงด้านความมั่นคงปลอดภัย และป้องกันการโจมตีที่อาจเกิดขึ้นในอนาคตได้อย่างมีประสิทธิภาพ

บรรณานุกรม

1. สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์. อะไรกันอยู่ดี ๆ ก็โดนแฮก ! องค์กรต้องรับมืออย่างไรในวันที่มิจฉาชีพก็ปรับตัวตามโลกดิจิทัล [เว็บไซต์]. สืบค้นเมื่อ 12 กรกฎาคม 2567, จาก https://www.etda.or.th/th/Useful-Resource/Knowledge-Sharing/Articles/Foresight/2_hacked.aspx.
2. สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.). สถิติภัยคุกคามทางไซเบอร์ [เว็บไซต์]. สืบค้นเมื่อ 31 พฤษภาคม 2567, จาก https://www.ncsa.or.th/service-statistics.html.
3. ไพบูลย์ พนัสบดี. OWASP Top 10 2021 – 10 อันดับต้องเช็คเพื่อเพิ่มความปลอดภัยให้เว็บแอปพลิเคชัน [เว็บไซต์]. สืบค้นเมื่อ 31 พฤษภาคม 2567, จาก https://www.cyfence.com/article/owasp-top-10-2021.
4. พิชญะ โมริโมโต. ลองทำ Vulnerability Assessment (VA) ในองค์กรด้วยตัวเองแบบง่ายๆ [เว็บไซต์]. สืบค้นเมื่อ 18 มิถุนายน 2567, จาก https://www.cyfence.com/article/how-to-va-pentest-in-your-organization/.
5. NT Cyfence. 10 แนวทางป้องกันภัยไซเบอร์ สำหรับธุรกิจขนาดกลางและเล็ก [เว็บไซต์]. สืบค้นเมื่อ 10 กรกฎาคม 2567, จาก https://www.cyfence.com/article/10-ways-to-prevent-cyber-threats-for-small-and-medium-businesses/.
6. Arponpong, C., & Soongpol, B. (2023, March). "Guideline Framework of Information Technology Security System by ISO 27001: 2013 Standard of the Securities and Exchange Commission (SEC)". Journal of Digital Business and Social Sciences, 9(1), JDB011, 1-12.
7. ACIS Professional Consulting Services offers a fully integrated information system. NIST Cybersecurity Framework 2.0 วางโครงสร้างองค์กรรับมือภัยไซเบอร์ [เว็บไซต์]. สืบค้นเมื่อ 4 กุมภาพันธ์ 2568, จาก https://www.acisonline.net/?p=11093.

บรรณานุกรม (ต่อ)

8. ณัชณภัทร ใจอดทน. การประเมินค่าช่องโหว่ของเว็บไซต์และการป้องกัน กรณีศึกษาเว็บไซต์กรมควบคุมการปฏิบัติทางอากาศ. สารนิพนธ์วิศวกรรมมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์, 2560.
9. OWASP. (n.d.). OWASP OpenVAS - OWASP Top 10 Security Risks. OWASP Foundation. Retrieved December 15, 2024, from https://owasp.org/Top10/.
10. M. U. Aksu, E. Altuncu, and K. Bicakci, "A first look at the usability of OpenVAS vulnerability scanner." in Workshop on Usable Security (USEC), San Diego, CA, USA, Feb. 2019. DOI: 10.14722/usec.2019.23026.
11. R. Kushe, "Comparative study of vulnerability scanning tools: Nessus vs Retina." International Scientific Journal "Security & Future", vol. 1, no. 2, pp. 69-71, 2017. [Online]. Available: https://stumejournals.com/journals/confsec/2017/2/69. Accessed: Aug. 1, 2024.
12. สมศักดิ์ เมาลีทอง. ระบบสารบรรณอิเล็กทรอนิกส์บนเครื่องบริการเว็บอย่างมั่นคงปลอดภัย. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช, 2564.
13. ร. ประสนิท, ศ. แซ่วื้อ, น. พนาวาส, และ จ. ชมยิ้ม, "การตรวจประเมินความสอดคล้องมาตรการรักษาความมั่นคงปลอดภัยสารสนเทศเทียบกับมาตรฐาน ISO/IEC 27001:2022 และการตรวจสอบช่องโหว่ทางเทคนิคของระบบปฏิบัติการบริษัท A." Journal of Science Innovation for Sustainable Development, vol. 5, no. 1, pp. 1-11, 2566.
14. ทัญญะ สิทธิมณีวรรณ. การพัฒนากรอบการจัดการช่องโหว่ระบบสารสนเทศเพื่อการรักษาความมั่นคงปลอดภัยทางไซเบอร์ กรณีศึกษา กองบัญชาการกองทัพภาคที่ 2. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช, 2564.

บรรณานุกรม (ต่อ)

15. นนทวัฒน์ ทับทิม สุปกรณ์ สวัสดิ์พุทรา. ระบบสแกนช่องโหว่ด้วยราสเบอร์รี่ไพน์. ปริญญานิพนธ์วิศวกรรมศาตรบัณฑิต ภาควิชาวิศกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2563.
16. วิชสุนี ธีรรัชต์กาญจน์. การสำรวจช่องโหว่เครือข่ายเพื่อการปฏิบัติงานด้านความปลอดภัยองค์กร. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2562.
17. สุมาศ กันจินะ. การศึกษาการโจมตี และแนวทางการป้องกัน Cross-Site Scripting (XSS). สารนิพนธ์วิทยาศาสตร์มหาบัณฑิต สาขาความมั่นคงทางระบบสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีมหานคร, 2560.
18. ปิยะ อาจหาญ. การโจมตีเซิร์ฟเวอร์ด้วยเอสคิวแอลอินเจ็คชั่นผ่านทีทีพีเฮดเดอร์ และแนวทางป้อง. สารนิพนธ์วิศวกรรมมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ และโทรคมนาคม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์, 2559.
19. Forum of Incident Response and Security Teams. (n.d.). Common Vulnerability Scoring System Version 3.0 Calculator. FIRST. https://www.first.org/cvss/calculator/3-0.

ประวัติผู้เขียน

ชื่อ	ธีระพงษ์ จำปาทอง
ชื่อการค้นคว้าอิสระ	การประเมินเครื่องมือสแกนช่องโหว่บนเว็บแอปพลิเคชันเพื่อพัฒนากลยุทธ์การป้องกันการโจมตีอย่างมีประสิทธิภาพ
สาขาวิชา	การบริหารเครือข่ายและความมั่นคงปลอดภัยสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ประวัติ	สำเร็จการศึกษาระดับปริญญาตรี สาขาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี พ.ศ. 2556 – ปัจจุบัน ทำงานบริษัท เทิร์นคีย์ คอมมูนิเคชั่น เซอร์วิส จำกัด (มหาชน)