EasyChair Preprint

№ 13118

Improving Web Server System Security with Automatic System Hardening

Aphinop Sangsilla and Pongsarun Boonyapakorn

April 27, 2024

การปรับปรุงความปลอดภัยระบบเว็บเซิร์ฟเวอร์ด้วยเครื่องมือแบบอัตโนมัติ
Improving Web Server System Security with Automatic System Hardening

อภินพ สังสิลลา (Aphinop Sangsilla)[1] และพงศ์ศรัณย์ บุญโญปกรณ์ (Pongsarun Boonyapakorn)[1]
[1]ภาควิชาการบริหารเครือข่ายดิจิทัลและความมั่นคงปลอดภัยสารสนเทศ คณะเทคโนโลยีสารสนเทศและนวัตกรรมดิจิทัล
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
S6507031857044@email.kmutnb.ac.th, pongsarun.b@itd.kmutnb.ac.th

บทคัดย่อ

การปรับปรุงความปลอดภัยระบบเว็บเซิร์ฟเวอร์ด้วยวิธี System Hardening เป็นหนึ่งในมาตรการที่สำคัญและมีประสิทธิภาพในการปกป้องระบบสารสนเทศจากการถูกโจมตีหรือการเข้าถึงโดยไม่ได้รับอนุญาต System Hardening เป็นกระบวนการที่ประกอบด้วยการปิดช่องโหว่ที่อาจเกิดขึ้น การลดฟังก์ชันการใช้งานที่ไม่จำเป็น และการยกระดับการรักษาความปลอดภัยของระบบภายในองค์กร โดยมีแนวทางปฏิบัติที่สำคัญ ได้แก่ การอัปเดตแพตช์และซอฟต์แวร์ให้เป็นรุ่นล่าสุด การปิดบริการที่ไม่จำเป็น และการจัดการสิทธิ์การเข้าถึงอย่างเข้มงวด แต่วิธีการทำแบบปกตินั้นผู้ทำต้องใช้เวลานานในการตั้งค่า เพราะมีแนวปฏิบัติหลายร้อยข้ออาจทำให้เกิดข้อผิดพลาดได้ง่าย ส่งผลให้ระบบเกิดช่องโหว่ได้ งานวิจัยนี้จึงมีวัตถุประสงค์เพื่อนำเครื่องมือแบบอัตโนมัติมาประยุกต์ใช้ในการทำ System Hardening เพื่อศึกษาประสิทธิภาพการใช้งานของเครื่องมือแบบอัตโนมัติในการลดจำนวนช่องโหว่ของระบบเว็บเซิร์ฟเวอร์

จากการวัดประสิทธิภาพโดยใช้การประเมินช่องโหว่เพื่อเปรียบเทียบจำนวนช่องโหว่ก่อนทำและหลังทำ System Hardening ด้วยเครื่องมืออัตโนมัติ ผลการวิจัยพบว่ามีประสิทธิภาพทั้งในเรื่องของการลดจำนวนของช่องโหว่ที่เป็นระดับความรุนแรง Critical และ High ได้ สามารถปรับปรุงการตั้งค่าด้านความปลอดภัยได้ตาม Security Baseline รวมถึงใช้เวลาน้อยลงกว่าแบบปกติ การลดจำนวนของช่องโหว่หลังการปรับปรุงเป็นหลักฐานที่ชัดเจนว่าเครื่องมือนั้นมีประสิทธิภาพในการป้องกันและลดความเสี่ยงจากการโจมตีทางไซเบอร์ได้ รวมถึงช่วยลดเวลาในการกำหนดค่า และช่วยลดข้อผิดพลาดของมนุษย์ได้

คำสำคัญ: การเสริมความแข็งแกร่งให้กับระบบ เครื่องมืออัตโนมัติ การประเมินช่องโหว่ ความมั่นคงปลอดภัยทางไซเบอร์

Abstract

Web server security enhancement through System Hardening is one of the critical and effective measures to protect information systems from attacks or unauthorized access. System Hardening is a process that involves closing potential vulnerabilities, reducing unnecessary functionality, and enhancing the security of the organization's internal systems. The key practices include updating patches and software to the latest versions, disabling unnecessary services, and implementing strict access control management. However, the traditional manual approach to System Hardening can be time-consuming and error-prone due to the numerous guidelines, which may result in system vulnerabilities. This research aims to apply an automated tool to perform System Hardening in order to study its effectiveness in reducing the number of vulnerabilities on web servers.

By measuring the performance through vulnerability assessment, this research compared the number of vulnerabilities before and after applying automated System Hardening tools. The results indicate that the automated approach was effective in reducing the number of critical and high-severity

vulnerabilities, while also being able to improve the security settings according to the recommended Security Baseline, and required less time compared to the manual process. The significant reduction in the number of vulnerabilities after the improvement serves as clear evidence that the tool is effective in preventing and mitigating the risks of cyber attacks. Furthermore, the automated approach can help reduce the time required for configuration and minimize human errors.

Keywords: System Hardening, Automation, Vulnerability Assessment, Cybersecurity.

1. บทนำ

ในปัจจุบันปัญหาภัยคุกคามทางไซเบอร์เกิดขึ้นกับองค์กรทั้งภาครัฐและเอกชน การละเมิดด้านความมั่นคงปลอดภัยทางไซเบอร์โดยใช้ประโยชน์จากช่องโหว่หรือจุดอ่อนต่างๆ ของระบบคอมพิวเตอร์นำไปสู่การรั่วไหลของข้อมูลสำคัญขององค์กรหรือทำให้ระบบเสียหาย เสียหาย ซึ่งเป้าหมายหลักของผู้โจมตี คือ ระบบเว็บเซิร์ฟเวอร์ เนื่องจากเป็นระบบที่อยู่บนอินเทอร์เน็ตจึงทำให้ผู้โจมตีเข้าถึงได้ง่ายและสามารถค้นหาช่องโหว่ได้ตลอดเวลา หากพบช่องโหว่หรือจุดอ่อนต่างๆ ก็จะสามารถทำการโจมตีผ่านช่องโหว่นั้นทันที ปัจจัยหนึ่งที่ทำให้เกิดช่องโหว่คือ ขาดการกำหนดค่าความปลอดภัยของระบบที่ดีเพียงพอ ซึ่งเป็นช่องโหว่หนึ่งใน OWASP Top 10 ปี 2021 เรื่อง A05 Security Misconfiguration ซึ่งวิธีหนึ่งที่จะสามารถช่วยลดความเสี่ยงเหล่านี้ได้คือ การเสริมความแข็งแกร่งให้กับระบบ (System Hardening) วิธีการทำ System Hardening แบบปกติ ผู้ดูแลระบบส่วนใหญ่อ้างอิง Security Baseline Hardening จากองค์กรที่มีความน่าเชื่อถือและได้รับการยอมรับ เช่น Center for Internet Security หรือ CIS ซึ่งมีข้อแนะนำในการปิดช่องโหว่จำนวนมากหลายร้อยข้อ ซึ่งอาจใช้เวลานานในการตั้งค่าและทำให้เกิดข้อผิดพลาดได้ง่าย ส่งผลทำให้ระบบเกิดช่องโหว่ได้

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาประสิทธิภาพของเครื่องมือแบบอัตโนมัตินำมาประยุกต์ใช้ในการทำ System Hardening ของระบบเว็บเซิร์ฟเวอร์ โดยใช้การประเมินช่องโหว่เพื่อเปรียบเทียบจำนวนช่องโหว่ก่อนทำและหลังทำ System Hardening เพื่อลดจำนวนช่องโหว่ด้านความปลอดภัยให้กับระบบ รวมถึงลดเวลาที่ใช้ในการตั้งค่า และช่วยลดข้อผิดพลาดของมนุษย์ สามารถนำไปประยุกต์ในองค์กรเพื่อลดความเสี่ยงจากการโจมตีทางไซเบอร์ได้

2. งานวิจัยที่เกี่ยวข้อง

ในปัจจุบันมีระบบเว็บเซิร์ฟเวอร์ที่นิยมใช้กันในปัจจุบัน 3 อันดับแรกคือ Apache HTTP Server, NGINX และ Microsoft IIS Web Server [1], จากความนิยมใช้งานสูงจึงเป็นเป้าหมายหลักของผู้โจมตีในการค้นหาช่องโหว่และเจาะระบบเว็บเซิร์ฟเวอร์ได้ [2] โดยเฉพาะช่องโหว่ที่มีระดับความรุนแรงสูงที่ทำให้สามารถเข้าถึงระบบได้ [3] ผู้วิจัยจึงเลือกเว็บเซิร์ฟเวอร์ดังกล่าวมาใช้ในการทำวิจัยนี้

System Hardening คือ กระบวนการป้องกันระบบเพื่อลดความเสี่ยงจากภัยคุกคามด้วยวิธีการต่าง ๆ เช่น การอัพเดทแพชท์ด้านความปลอดภัยของระบบปฏิบัติการหรือแอพพลิเคชัน, การปิดพอร์ตที่ไม่ใช้งาน เป็นต้น เพื่อลดช่องโหว่ต่าง ๆ ซึ่งส่วนใหญ่จะอ้างอิงแนวปฏิบัติพื้นฐาน (Security Baseline Hardening) จากองค์กรที่เป็นที่ยอมรับและน่าเชื่อถือ เช่น Center for Internet Security (CIS) [4]

Security Baseline Hardening ที่ใช้ในงานวิจัยนี้ได้อ้างอิงให้สอดคล้องกับ OWASP Top 10 ปี 2021 โดยเน้นไปที่ข้อ A05:2021 Security Misconfiguration [5], เป็นเรื่องการกำหนดค่าความปลอดภัยไม่ถูกต้อง ซึ่งสาเหตุที่อาจจะทำให้แอพพลิเคชันเว็บมีช่องโหว่ เช่น ขาดการกำหนดค่าด้านความปลอดภัย (Security Hardening) อย่างเหมาะสมทั้งในส่วนของแอพพลิเคชันเว็บ หรือระบบปฏิบัติการ ระบบขาดการอัพเดทแพชต์ด้านความปลอดภัยให้ทันสมัยหรือใช้ส่วนประกอบซอฟต์แวร์ที่ล้าสมัยหรือมีช่องโหว่ เป็นต้น

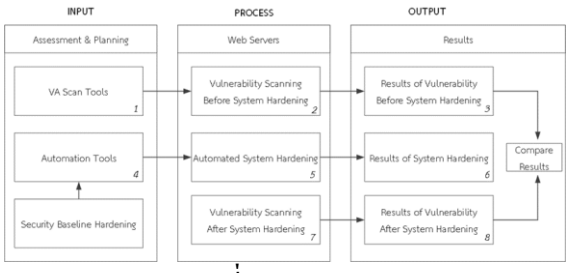
เครื่องมือแบบอัตโนมัติในปัจจุบันมีหลายเครื่องมือที่สามารถนำมาประยุกต์ใช้ โดยที่ผู้วิจัยสนใจใช้คือ Jenkins [6], Ansible [7], Chef Infra [8], และ Microsoft Security Compliance Tool Kit [9]

การประเมินช่องโหว่ คือ กระบวนการค้นหา ระบุช่องโหว่หรือจุดอ่อน การวิเคราะห์ผลลัพธ์ การประเมินความเสี่ยง และการแก้ไขช่องโหว่ของระบบ ซึ่งเครื่องมือที่ใช้ในการสแกนหาช่องโหว่ที่มีอยู่ในปัจจุบันและได้นำมาใช้ในงานวิจัย คือ Nessus [10], และ OpenVAS [11]

การเฝ้าระวังเหตุการณ์ภัยคุกคามทางไซเบอร์ให้กับระบบอย่างต่อเนื่อง โดยผู้วิจัยสนใจใช้โปรแกรม Wazuh [12] ที่มีความสามารถในการรวบรวมข้อมูลจราจรทางคอมพิวเตอร์จากเครื่องแม่ข่าย และนำมาวิเคราะห์ตรวจสอบแบบ Real-time เพื่อสามารถตอบสนองต่อเหตุการณ์ได้ทันท่วงที [13]

3. วิธีดำเนินการวิจัย

งานวิจัยนี้ศึกษาประเด็นหลักที่สนใจ คือ เรื่องจำนวนช่องโหว่ที่พบบนเครื่องเว็บเซิร์ฟเวอร์ก่อนการทำ System Hardening และจำนวนช่องโหว่ที่พบบนเครื่องเว็บเซิร์ฟเวอร์หลังการทำ System Hardening เพื่อศึกษาว่าการใช้เครื่องมือแบบอัตโนมัติจะช่วยลดจำนวนช่องโหว่หรือไม่ ซึ่งประกอบด้วย 5 ขั้นตอน คือ 1. การออกแบบเครื่องเว็บเซิร์ฟเวอร์เป้าหมาย 2. การเลือกเครื่องมือที่ใช้ในงานวิจัย 3. การประเมินช่องโหว่ความปลอดภัย (VA Scan) ก่อนการทำ System Hardening 4. การใช้เครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัติ 5. การประเมินช่องโหว่ความปลอดภัย (VA Scan) หลังการทำ System Hardening โดยมีกระบวนการตามภาพที่ 1



ภาพที่ 1: System Diagram

3.1 การออกแบบเครื่องเว็บเซิร์ฟเวอร์เป้าหมาย

การออกแบบเครื่องเว็บเซิร์ฟเวอร์เป้าหมายโดยการสร้างระบบคอมพิวเตอร์เสมือน (Virtual Machine) ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์เป้าหมายที่จะดำเนินการประเมินช่องโหว่ (Vulnerability Assessment) และทำ System Hardening ตามตารางที่ 1

ตารางที่ 1: เครื่องเว็บเซิร์ฟเวอร์เป้าหมาย

ลำดับ	*ระบบปฏิบัติการ*	*เว็บเซิร์ฟเวอร์*	*แอพพลิเคชันเว็บ*
1	RedHat Linux 8	Apache	WordPress
2	RedHat Linux 8	NGINX	WordPress
3	Windows 2022	IIS	WordPress
4	Ubuntu 22	Apache	WordPress

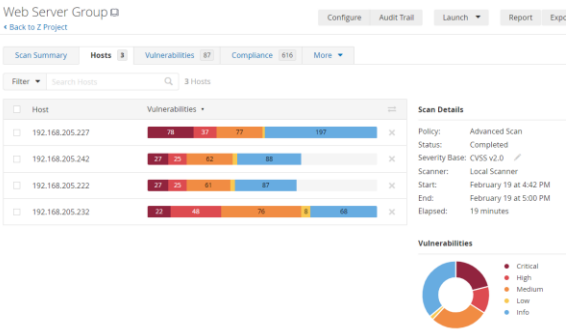
3.2 การเลือกเครื่องมือที่ใช้ในงานวิจัย

การเลือกเครื่องมือที่ใช้ในงานวิจัยแบ่งได้เป็น 3 ส่วน คือ 1. เครื่องมือประเมินช่องโหว่ความปลอดภัย (VA Scan) 2. เครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัติ และ 3. เครื่องมือเฝ้าระวังเหตุการณ์ด้านความปลอดภัยของเครื่องแม่ข่าย ตามตารางที่ 2

ลำดับ	*ประเภทเครื่องมือ*	*เครื่องมือ*
1	เครื่องมือประเมินช่องโหว่ความปลอดภัย	• Nessus • OpenVAS
2	เครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัติ	• Jenkins • Ansible • Chef Infra • Security Compliance Toolkit
3	เครื่องมือเฝ้าระวังเหตุการณ์ด้านความปลอดภัยของเครื่องแม่ข่าย	Wazuh

3.3 การประเมินช่องโหว่ความปลอดภัย (VA Scan) ก่อนการทำ System Hardening

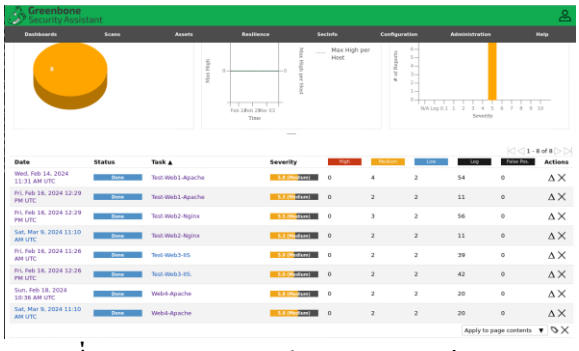
การประเมินช่องโหว่ความปลอดภัย (VA Scan) โดยใช้เครื่องมือ VA Scan บนเครื่องเว็บเซิร์ฟเวอร์ก่อนการทำ System Hardening เพื่อเก็บรวบรวมข้อมูลสถิติว่ามีจำนวนของช่องโหว่เท่าไร จากภาพที่ 2 แสดงจำนวนช่องโหว่ของเครื่องเว็บเซิร์ฟเวอร์โดยการ VA Scan ด้วยเครื่องมือ Nessus ก่อนการทำ System Hardening



ภาพที่ 2: ผลการ VA Scan โดย Nessus ก่อนการทำ

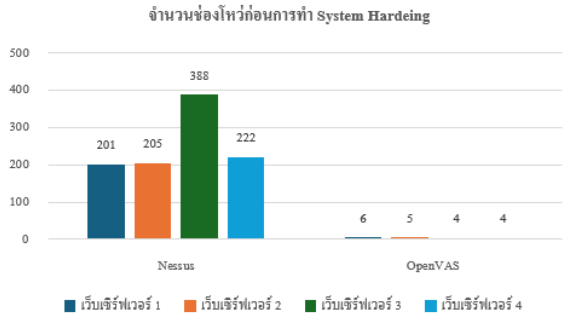
System Hardening เครื่องเว็บเซิร์ฟเวอร์

จากภาพที่ 3 เป็นการแสดงจำนวนช่องโหว่ของเครื่องเว็บเซิร์ฟเวอร์โดยการ VA Scan ด้วยเครื่องมือ OpenVAS ก่อนการทำ System Hardening



ภาพที่ 3: ผลการ VA Scan โดย OpenVAS ก่อนการทำ System Hardening เครื่องเว็บเซิร์ฟเวอร์

จากภาพที่ 4 เป็นการเปรียบเทียบจำนวนช่องโหว่ก่อนการทำ System Hardening ของเว็บเซิร์ฟเวอร์ โดยการใช้เครื่องมือ Nessus กับ OpenVAS ผู้วิจัยพบว่า Nessus นั้นสามารถค้นหาช่องโหว่ได้ดีกว่า OpenVAS



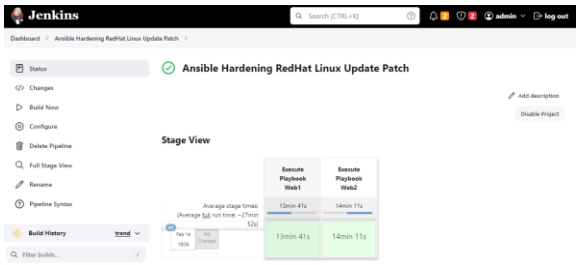
ภาพที่ 4: จำนวนช่องโหว่ก่อนการทำ System Hardening

3.4 การใช้เครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัติ

การใช้เครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัติในการทำ System Hardening โดยอ้างอิงสาเหตุที่ทำให้เกิดช่องโหว่ของระบบเว็บเซิร์ฟเวอร์จาก OWASP Top 10 ปี 2021 ข้อ A05 Security Misconfiguration เป็นเรื่องของการกำหนดค่าความปลอดภัยไม่ถูกต้อง จากนั้นจึงใช้เป็นแนวทางปฏิบัติโดยอ้างอิง Security Baseline Hardening จากแหล่งที่น่าเชื่อถือ เช่น CIS Benchmarks ดังตารางที่ 3

OWASP Top 10 A05:2021 Security Misconfiguration	*แนวทางปฏิบัติ Security Baseline Hardening*
1. ขาดการกำหนดค่าความปลอดภัย (security hardening) อย่างเหมาะสมในส่วนของระบบแอพพลิเคชันเว็บ หรือส่วนของระบบปฏิบัติการ เช่น เว็บเซิร์ฟเวอร์เปิดใช้งาน Directory listening อาจจะทำให้ผู้โจมตีสามารถอ่านหรือดาวน์โหลดไฟล์สำคัญผ่านทางหน้าเว็บเบราว์เซอร์ได้ หรือเว็บเซิร์ฟเวอร์เปิดเผยข้อมูลซึ่งอาจจะมีความละเอียดอ่อนของระบบ อาจจะทำให้ผู้โจมตีสามารถใช้ประโยชน์ในการโจมตีได้	• กำหนดค่าเรื่อง Information Leakage ไม่เปิดเผยข้อมูลเกี่ยวกับเวอร์ชั่นของระบบแอพพลิเคชันเว็บที่ใช้งาน • ปิดการใช้งาน Directory listening บนเครื่องเว็บเซิร์ฟเวอร์เพื่อป้องกันไม่ให้ผู้โจมตีสามารถเข้าถึงข้อมูลสำคัญผ่านทางหน้าเว็บเบราว์เซอร์ได้ • กำหนดค่าส่วนหัวให้มีความปลอดภัย (Security Headers)
2.ระบบขาดการอัปเดทแพตช์ความปลอดภัยให้ทันสมัยหรือใช้ส่วนประกอบซอฟต์แวร์ที่ล้าสมัยหรือมีช่องโหว่	• ทำการอัปเดทแพตช์เกี่ยวกับเรื่องของด้านความปลอดภัยของระบบปฏิบัติการและแอพพลิเคชันเว็บให้ทันสมัย
3. บัญชีผู้ใช้เริ่มต้นและรหัสผ่านที่ยังคงเปิดใช้งานและไม่ทำการเปลี่ยนแปลงหรือไม่มีการควบคุมการใช้งานบัญชีสิทธิพิเศษ	• ปิดการใช้งานบัญชีผู้ใช้เริ่มต้นหรือเปลี่ยนรหัสผ่าน • จำกัดการเข้าถึงจากระยะไกลของบัญชีสิทธิพิเศษ

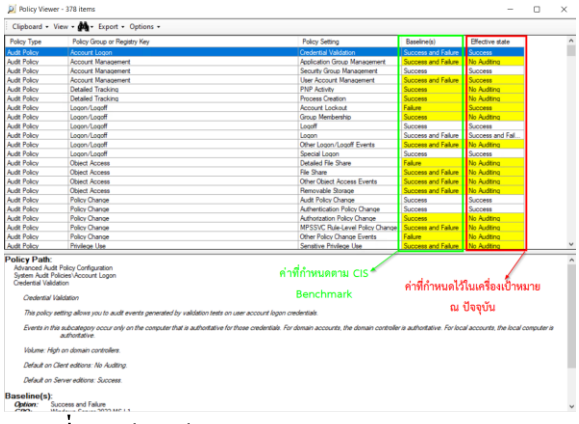
การใช้เครื่องมืออัตโนมัติ Jenkins ทำงานร่วมกับ Ansible โดยใช้ภาษา YAML ในการเขียนชุดคำสั่งสคริปต์ Hardening ให้ทำงานแบบอัตโนมัติผ่านทางหน้าจอเว็บ GUI (Graphic User Interface) ช่วยให้ผู้ใช้งานเข้าใจง่าย ซึ่งสามารถสั่งให้ทำงานไปยังเครื่องเป้าหมายได้หลายๆ เครื่องพร้อมกัน และสามารถติดตามผลการทำงานได้ ดังภาพที่ 5



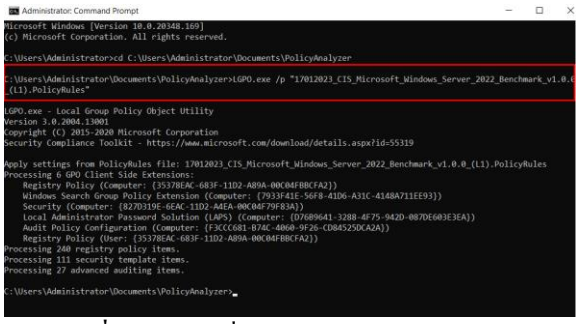
ภาพที่ 5: หน้าจอแสดงการทำงานของ Jenkins

จากภาพที่ 6 ถึง 7 การใช้ชุดเครื่องมืออัตโนมัติ Microsoft Security Compliance Tool Kit สามารถเทียบค่า Group Policy File และ Registry Value มาเทียบกับ Configuration Template โดยจะอ้างอิงตาม CIS Microsoft Windows Server 2022 Benchmark และทำการส่งค่าไปยัง

เครื่องเป้าหมาย ซึ่งพบว่าสามารถทำงานได้สำเร็จโดยใช้เวลาประมาณ 1 นาที



ภาพที่ 6: เปรียบเทียบค่า Windows configuration ก่อนทำการ System Hardening



ภาพที่ 7: การใช้เครื่องมืออัตโนมัติในการ System Hardening บนระบบ Windows

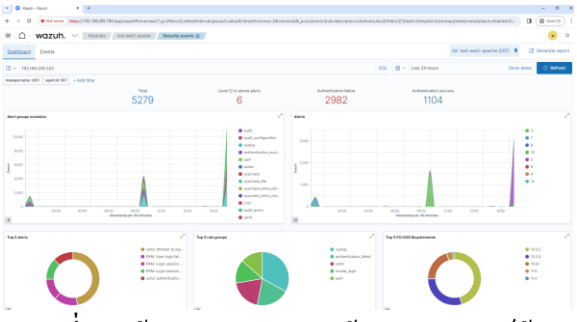
จากภาพที่ 8 การใช้ชุดเครื่องมืออัตโนมัติ Chef Infra ใช้ภาษา Ruby ในการเขียนชุดคำสั่งเรียกว่า Cookbooks ให้ทำ Hardening แบบอัตโนมัติ

ภาพที่ 8: ภาพคำสั่ง Cookbooks

3.5 การประเมินช่องโหว่ความปลอดภัย (VA Scan) หลังการทำ System Hardening

ทำการเปรียบเทียบจำนวนช่องโหว่ก่อนทำและหลังทำ System Hardening เพื่อศึกษาประสิทธิภาพของเครื่องมือตั้งค่าความปลอดภัยแบบอัตโนมัตินั้นสามารถลดจำนวนช่องโหว่ได้หรือไม่

ขั้นตอนการเฝ้าระวังเหตุการณ์ด้านความปลอดภัยของเครื่องแม่ข่ายเว็บเซิร์ฟเวอร์ โดยใช้โปรแกรม Wazuh ในการติดตามเฝ้าระวังเหตุการณ์ด้านความปลอดภัยอย่างต่อเนื่อง โดยการเก็บข้อมูลจราจรทางคอมพิวเตอร์จากเครื่องแม่ข่าย และนำมาวิเคราะห์ตรวจสอบแบบ Real-time มาแสดงบนหน้า Dashboard ที่รวบรวมเหตุการณ์แจ้งเตือน (Alert) เรื่องต่างๆ เพื่อให้สามารถตอบสนองต่อเหตุการณ์ภัยคุกคามได้ทันที ดังภาพที่ 9

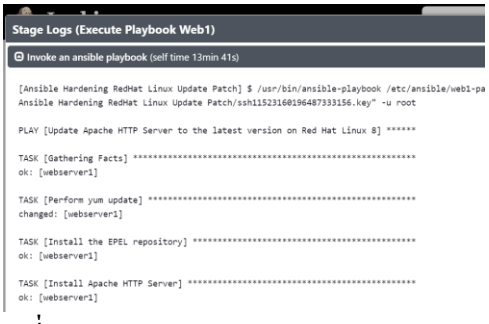


ภาพที่ 9: หน้า Dashboard ระบบเฝ้าระวังเหตุการณ์ด้านความปลอดภัย

4. ผลการดำเนินงาน

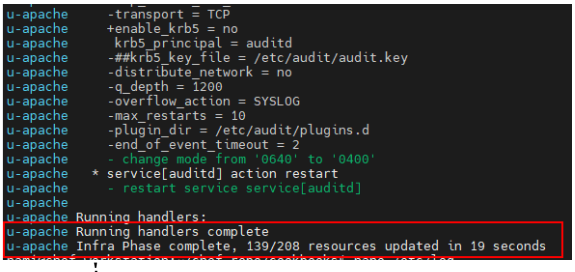
4.1 ผลการทำ System Hardening ด้วยเครื่องมือแบบอัตโนมัติ

จากภาพที่ 10 เป็นผลการทำ System Hardening ด้วยเครื่องมือ Jenkins ซึ่งพบว่าสามารถสั่งให้ Ansible ทำการการอัปเดตแพตช์และซอฟต์แวร์ให้เป็นรุ่นล่าสุดได้สำเร็จ ซึ่งใช้เวลาประมาณ 13 นาที

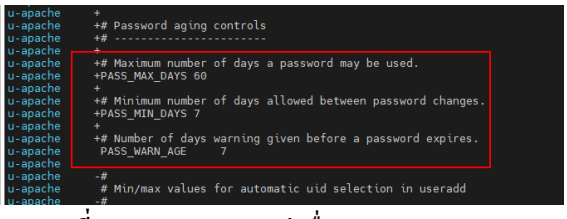


ภาพที่ 10: ผลการทำ System Hardening ของ Ansible

จากภาพที่ 11 เป็นผลการทำ System Hardening ด้วยเครื่องมือ Chef ซึ่งพบว่าสามารถสั่งให้ทำงานตามที่ได้กำหนดไว้สำเร็จซึ่งใช้เวลาประมาณ 1 นาที ซึ่งเป็นเรื่องการกำหนดค่า Password Policy ให้ปลอดภัยตามภาพที่ 12

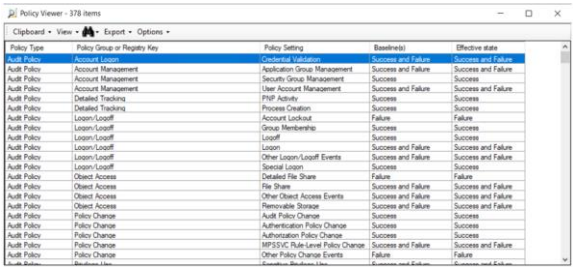


ภาพที่ 11: ผลการทำ System Hardening ของ Chef



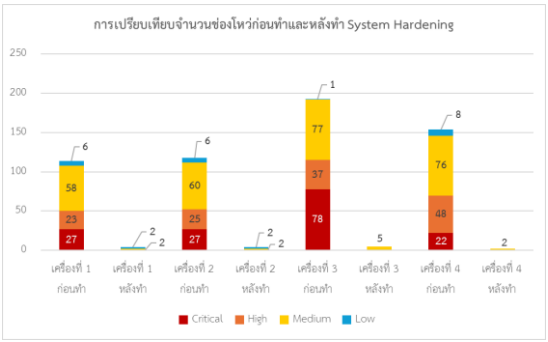
ภาพที่ 12: ผลการกำหนดค่าเรื่อง Password Policy

จากภาพที่ 13 เป็นผลการทำ System Hardening ด้วยเครื่องมือ Microsoft Security Compliance Tool Kit ซึ่งพบว่ามีค่าตรงตาม Security Baseline เช่น เรื่องการบันทึกกิจกรรมต่างๆ ในการเข้าถึงเครื่องเซิร์ฟเวอร์ เป็นต้น



ภาพที่ 13: การเปรียบเทียบค่า Windows configuration หลังการทำ System Hardening

4.2 ผลการเปรียบเทียบจำนวนช่องโหว่ก่อนทำและหลังการทำ System Hardening ด้วยเครื่องมืออัตโนมัติ



ภาพที่ 14: เปรียบเทียบจำนวนช่องโหว่เว็บเซิร์ฟเวอร์

5. สรุปผล

การปรับปรุงการตั้งค่าด้านความปลอดภัยได้ตาม Security Baseline ของระบบเว็บเซิร์ฟเวอร์ และการลดจำนวนของช่องโหว่โดยเฉพาะช่องโหว่ที่เป็นระดับความรุนแรง Critical และ High เป็นหลักฐานที่ชัดเจนว่าเครื่องมือนั้นมีประสิทธิภาพในการป้องกันและลดความเสี่ยงจากการโจมตีทางไซเบอร์ รวมถึงการใช้เวลาในการกำหนดค่า System Hardening น้อยกว่าแบบปกติที่ต้องใช้เวลานานหลายชั่วโมง สามารถช่วยลดเวลาในการกำหนดค่าเหลือเพียงหลักนาที รวมถึงสามารถช่วยลดข้อผิดพลาดของมนุษย์ได้

เอกสารอ้างอิง

[1] Top Web Server Technologies. จาก: https://www.similartech.com/categories/web-server

[2] Ronald Clack. “Security Automation for Windows 10 Hosts: Hardening of Windows 10 Password Policy,” *Master’s thesis*. Information and Communication Technology, Master of Engineering, Cyber Security, School of Technology. 2020.

[3] พัชรวัฒน์ โกสิตงามดีวงศ์., “การประเมินช่องโหว่ระบบบริหารจัดการทางการแพทย์ กรณีศึกษา โรงพยาบาลภูมิพลอดุลยเดชกรมแพทย์ทหารอากาศ” *สารนิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต* สาขาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมด้านเทคโนโลยี และวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

[4] The Center for Internet Security, Inc. (CIS). จาก: https://www.cisecurity.org/about-us

[5] OWASP Top 10 From: https://owasp.org/Top10/A05_2021-Security_Misconfiguration.

[6] Jenkins. จาก: https://Jenkinss.io

[7] Ansible. จาก: https://www.ansible.com

[8] Chef Infra. จาก: https://www.chef.io

[9] Microsoft Security Compliance Toolkit and Baselines. จาก: https://www.microsoft.com/en-us/download/details.aspx?id=55319

[10] Tenable Nessus Vulnerability Assessment Scan Tool จาก: https://www.tenable.com

[11] OpenVAS. From: https://www.openvas.org

[12] Wazuh The Open Source Security Platform. จาก: https://wazuh.com

[13] ปิยะนันต์ จำนงสุทธเสถียร และพิชิต ลีรุ่งนาวารัตน์., “การพัฒนากระบวนการจัดการข้อมูลด้านความปลอดภัย และเหตุการณ์ ด้านความปลอดภัยของเครื่องแม่ข่าย,” *วารสาร ปข มท*,. ปีที่ 12, ฉบับที่ 2, พ.ค.-ส.ค. 2566 หน้า 163 - 172