ประกาศมหาวิทยาลัยพะเยา

เรื่อง แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์
และเว็บแอปพลิเคชัน มหาวิทยาลัยพะเยา พ.ศ. 2568

โดยที่เป็นการสมควรให้มีประกาศมหาวิทยาลัยพะเยา เรื่อง แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน มหาวิทยาลัยพะเยา พ.ศ. 2568 เพื่อเป็นแนวปฏิบัติให้ส่วนงานในการให้บริการเว็บไซต์และเว็บแอปพลิเคชันของส่วนงาน เป็นไปอย่างมีประสิทธิภาพสอดคล้องกับการพัฒนาระบบมาตรฐานการจัดการความมั่นคงปลอดภัยสารสนเทศ ISO/IEC 27001:2022 ยกระดับการบริหารจัดการด้านความมั่นคงปลอดภัยไซเบอร์ ความมั่นคงปลอดภัยของข้อมูล และการคุ้มครองข้อมูลส่วนบุคคล รวมทั้งป้องกัน รับมือและลดความเสี่ยงจากภัยคุกคามทางไซเบอร์ สร้างความเชื่อใจแก่ผู้รับบริการ จึงได้กำหนดแนวปฏิบัติตามมาตรฐาน ISO/IEC 27001:2022 และแนวปฏิบัติทั้งในระดับสากลและที่จัดทำขึ้นโดยหน่วยงานภายในประเทศ ได้แก่ NIST 800-44 v2: Guidelines on Securing Publix Web Servers, OWASP Top Ten 2021, Mozilla Web Security และแนวปฏิบัติการรักษาความมั่นคงปลอดภัยเว็บไซต์ (Website Security Guideline) โดย สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)

อาศัยอำนาจตามความในมาตรา 33 แห่งพระราชบัญญัติมหาวิทยาลัยพะเยา พ.ศ. 2553 ประกอบกับมติคณะกรรมการบริหารมหาวิทยาลัยพะเยา ในคราวประชุมครั้งที่ 150 (13/2568) เมื่อวันที่ 18 กรกฎาคม 2568 จึงออกประกาศไว้ดังนี้

ข้อ 1 ประกาศนี้ เรียกว่า “ประกาศมหาวิทยาลัยพะเยา เรื่อง แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน มหาวิทยาลัยพะเยา พ.ศ. 2568”

ข้อ 2 ประกาศนี้ให้ใช้บังคับตั้งแต่วันประกาศเป็นต้นไป

ข้อ 3 ในประกาศนี้

“มหาวิทยาลัย” หมายความว่า มหาวิทยาลัยพะเยา

“อธิการบดี” หมายความว่า อธิการบดีมหาวิทยาลัยพะเยา

“ส่วนงาน” หมายความว่า ส่วนงานตามมาตรา 7 แห่งพระราชบัญญัติมหาวิทยาลัยพะเยา พ.ศ. 2553 และให้หมายความรวมถึงหน่วยงานภายในส่วนงานตามมาตรา 7 (2) แห่งพระราชบัญญัติมหาวิทยาลัยพะเยา พ.ศ. 2553 หน่วยตรวจสอบภายใน และศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์

“เครื่องคอมพิวเตอร์แม่ข่าย” หมายความว่า เครื่องคอมพิวเตอร์แม่ข่ายหรือเครื่องคอมพิวเตอร์แม่ข่ายเสมือน (Virtual Machine) ที่ได้รับการติดตั้งซอฟต์แวร์และบริการที่ใช้สำหรับการให้บริการเว็บไซต์และ/หรือเว็บแอปพลิเคชัน”

ข้อ 4 ให้หัวหน้า...

ข้อ 4 ให้หัวหน้าส่วนงานเป็นผู้รับผิดชอบในการกำกับดูแลการดำเนินงานของส่วนงานให้สอดคล้องกับคู่มือแนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน มหาวิทยาลัย

ข้อ 5 รายละเอียดคู่มือแนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน มหาวิทยาลัย ให้เป็นไปตามเอกสารแนบท้ายประกาศนี้

ข้อ 6 ให้อธิการบดีรักษาการตามประกาศนี้ กรณีที่มีปัญหาเกี่ยวกับการบังคับใช้หรือการปฏิบัติตามประกาศนี้ อธิการบดีมีอำนาจตีความและวินิจฉัยชี้ขาด การตีความและการวินิจฉัยให้ถือเป็นที่สุด

ประกาศ ณ วันที่ 31 กรกฎาคม พ.ศ. 2568

(รองศาสตราจารย์ ดร.สุภกร พงศบางโพธิ์)
อธิการบดีมหาวิทยาลัยพะเยา

เอกสารแนบท้าย

ประกาศมหาวิทยาลัยพะเยา เรื่อง แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์
และเว็บแอปพลิเคชัน มหาวิทยาลัยพะเยา พ.ศ. 2568

คู่มือ

แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่าย เว็บไซต์และเว็บแอปพลิเคชัน มหาวิทยาลัยพะเยา

เวอร์ชัน 1.0

โดย
ศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์
และ
ศูนย์บริการเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยพะเยา

สารบัญ

บทนำ

หลักการและเหตุผล

ในยุคดิจิทัลปัจจุบัน การใช้งานเว็บไซต์และบริการออนไลน์มีความสำคัญอย่างยิ่งต่อการดำเนินธุรกิจและการให้บริการต่าง ๆ อย่างไรก็ตาม การเพิ่มขึ้นของการใช้งานเทคโนโลยีสารสนเทศยังนำมาซึ่งความเสี่ยงด้านความมั่นคงปลอดภัยที่สูงขึ้น การโจมตีทางไซเบอร์ เช่น การแฮ็กข้อมูล การโจมตีแบบ DDoS และการขโมยข้อมูลส่วนบุคคล เป็นภัยคุกคามที่สามารถเกิดขึ้นได้ตลอดเวลา

เพื่อป้องกันและลดความเสี่ยงจากภัยคุกคามเหล่านี้ การจัดทำแนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน จึงเป็นสิ่งจำเป็นที่จะช่วยให้ผู้ดูแลระบบสามารถปฏิบัติตามมาตรฐานความมั่นคงปลอดภัยที่กำหนดไว้ได้อย่างมีประสิทธิภาพ สามารถรับมือกับปัญหาที่อาจเกิดขึ้นได้อย่างเหมาะสม และสร้างความมั่นใจให้กับผู้ใช้งานว่าระบบมีความปลอดภัยและสามารถป้องกันภัยคุกคามต่าง ๆ ได้อย่างมีประสิทธิภาพ

วัตถุประสงค์

1. เพื่อให้แน่ใจว่าข้อมูลที่เก็บอยู่ในเครื่องคอมพิวเตอร์แม่ข่ายจะไม่ถูกเข้าถึงหรือแก้ไขโดยผู้ที่ไม่ได้รับอนุญาต
2. เพื่อป้องกันการแก้ไขหรือทำลายข้อมูลที่สำคัญ
3. เพื่อป้องกันการโจมตีที่อาจทำให้ระบบหยุดทำงานหรือข้อมูลถูกขโมย
4. เพื่อให้ผู้ใช้งานมั่นใจในความปลอดภัยของระบบและบริการที่ใช้งาน
5. เพื่อให้การดำเนินงานเป็นไปตามกฎหมายและมาตรฐานความมั่นคงปลอดภัยที่กำหนดไว้

การจัดทำแนวปฏิบัติ

แนวปฏิบัติฉบับนี้จัดทำขึ้นโดยใช้เนื้อหาอ้างอิงมาจากมาตรฐาน เอกสารเผยแพร่ และแนวปฏิบัติทั้งที่เป็นสากลและที่จัดทำขึ้นโดยหน่วยงานภายในประเทศไทย โดยมีแหล่งอ้างอิงดังนี้

1. NIST 800-44 v2 Guidelines on Securing Public Web Servers
2. OWASP Top Ten 2021
3. Mozilla Web Security
4. แนวปฏิบัติในการรักษาความมั่นคงปลอดภัยไซเบอร์เว็บไซต์ (Website Security Guideline) โดย สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)
5. รายงานสรุปโครงการระบบตรวจจับพฤติกรรมและวิเคราะห์ข้อมูลอาชญากรรมทางไซเบอร์เชิงรุกสำหรับหน่วยงานโครงสร้างพื้นฐานสำคัญทางสารสนเทศและหน่วยงานภาครัฐ (Proactive Cyber Watch) โดย สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ

แนวปฏิบัติความมั่นคงปลอดภัยเครื่องคอมพิวเตอร์แม่ข่ายเว็บไซต์และเว็บแอปพลิเคชัน

1. ด้าน Secure Infrastructure

1.1 ปิดบริการที่ไม่ปลอดภัย หรือไม่จำเป็นต่อการให้บริการเว็บไซต์และเว็บแอปพลิเคชัน ดังนี้

การปิดพอร์ต (port) ที่ไม่มีการใช้งานและพอร์ตให้บริการที่ไม่ปลอดภัยบนเครื่องคอมพิวเตอร์แม่ข่ายจะช่วยลดช่องโหว่ที่อาจถูกโจมตีได้ โดยจำกัดจุดเข้าถึงสำหรับผู้ไม่หวังดี เพื่อเสริมความแข็งแกร่งด้านความปลอดภัยโดยรวมของระบบ โดยมีบริการที่ไม่ควรเปิดใช้งานดังนี้

(1) FTP (File Transfer Protocol)

(2) SMB (Server Message Block)

(3) Telnet

(4) PPTP (Point-to-Point Tunneling Protocol)

(5) SSH ที่เข้ารหัสไม่ปลอดภัย เช่น Arcfour หรือ Cipher Block Chaining (CBC)

(6) กำหนดค่า Firewall ให้เปิดใช้งาน port 443 เท่านั้น

หากมีความจำเป็นต้องเปิดใช้งานบริการหรือพอร์ตอื่น ๆ สามารถแจ้งศูนย์บริการเทคโนโลยีสารสนเทศและการสื่อสาร เพื่อดำเนินการเปิดใช้งาน และให้บันทึกข้อมูลวันเวลาที่เปิดใช้งาน

1.2 จำกัดการติดตั้งซอฟต์แวร์ที่ไม่จำเป็น หรือซอฟต์แวร์ที่ไม่ปลอดภัย

การจำกัดการติดตั้งเฉพาะซอฟต์แวร์ที่จำเป็นและผ่านการตรวจสอบแล้วบนเครื่องคอมพิวเตอร์แม่ข่ายจะช่วยลดความเสี่ยงจากการมีช่องโหว่ที่เกิดจากซอฟต์แวร์ที่ไม่ปลอดภัยหรือไม่ได้รับการอัปเดต ช่วยให้แน่ใจว่าระบบปฏิบัติการและแอปพลิเคชันที่ติดตั้งได้รับการปกป้องอย่างเหมาะสม โดยมีแนวปฏิบัติดังนี้

(1) ไม่ติดตั้งซอฟต์แวร์ที่ไม่จำเป็น หรือไม่เกี่ยวข้องกับการให้บริการ เช่น ซอฟต์แวร์สำนักงาน หรือซอฟต์แวร์สำหรับใช้งานในเครื่องคอมพิวเตอร์ลูกข่าย เป็นต้น

(2) ไม่ติดตั้งซอฟต์แวร์ เช่น crack tool ซอฟต์แวร์ที่มี malicious code และซอฟต์แวร์ละเมิดลิขสิทธิ์

(3) ตรวจสอบหรือรับการแจ้งเตือนจากแหล่งต่าง ๆ เกี่ยวกำหนดเวลาสิ้นสุดหรือวันหมดอายุการสนับสนุนของระบบปฏิบัติการหรือซอฟต์แวร์อื่น ๆ ที่ใช้งานอยู่ และจัดทำแผนการปรับปรุงระบบ เพื่อให้รองรับการใช้งานระบบปฏิบัติการหรือซอฟต์แวร์ที่ใหม่กว่า

1.3 ติดตั้งซอฟต์แวร์สำหรับตรวจจับและตอบสนองต่อภัยคุกคาม (Endpoint Detection and Response) และซอฟต์แวร์ตรวจจับป้องกันไวรัสคอมพิวเตอร์ (Anti-Virus Software)

ติดตั้ง Endpoint Detection and Response (EDR) เพื่อตรวจจับและตอบสนองต่อภัยคุกคามขั้นสูงแบบเรียลไทม์ ควบคู่ไปกับ Anti-Virus เพื่อป้องกันมัลแวร์ที่รู้จัก การทำงานร่วมกันของสองระบบนี้ช่วยเสริมสร้างการป้องกันภัยคุกคามทางไซเบอร์ได้อย่างครอบคลุม โดยมีแนวปฏิบัติดังนี้

(1) ติดตั้งหรือเปิดใช้งานซอฟต์แวร์ป้องกันไวรัสคอมพิวเตอร์ เช่น Windows Defender หรือซอฟต์แวร์อื่น ๆ ที่มหาวิทยาลัยมีให้บริการ

(2) เปิดการอัปเดตและป้องกันอัตโนมัติในส่วนของซอฟต์แวร์ป้องกันไวรัสคอมพิวเตอร์

(3) ติดตั้งซอฟต์แวร์ตรวจจับและตอบสนองต่อภัยคุกคาม เช่น Microsoft Endpoint Defender หรือซอฟต์แวร์อื่นๆ ที่มหาวิทยาลัยมีให้บริการ

1.4 อัปเดตแพตช์ซอฟต์แวร์ระบบปฏิบัติการและซอฟต์แวร์บริการอื่น ๆ ที่ใช้ให้เป็นปัจจุบันอยู่เสมอ

การอัปเดตแพตช์ความปลอดภัยสำหรับระบบปฏิบัติการและซอฟต์แวร์บริการอื่น ๆ อย่างสม่ำเสมอเป็นสิ่งสำคัญอย่างยิ่ง เพื่อแก้ไขช่องโหว่ที่ค้นพบใหม่ ป้องกันการโจมตีจากมัลแวร์และผู้ไม่ประสงค์ดี เสริมสร้างความมั่นคงปลอดภัยโดยรวมของเครื่องคอมพิวเตอร์แม่ข่าย โดยมีแนวปฏิบัติดังนี้

(1) เปิดการใช้งานการอัปเดตอัตโนมัติสำหรับ Windows Server ในส่วนของการอัปเดตที่มีความสำคัญสูง หรือเป็นการอัปเดตความปลอดภัยของระบบปฏิบัติการและซอฟต์แวร์อื่นๆ

(2) ตรวจสอบการอัปเดตเป็นอย่างสม่ำเสมอ เพื่อให้มั่นใจว่ามีการอัปเดตอย่างต่อเนื่อง และในบางกรณีอาจจำเป็นต้องมีการ restart เครื่อง เพื่อให้การอัปเดตถูกนำไปประยุกต์ใช้อย่างสมบูรณ์

(3) หากได้รับการแจ้งเตือนการอัปเดตด้านความปลอดภัยจากศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์ หรือข้อมูลภายนอกจากแหล่งที่มีความน่าเชื่อถือ ให้ผู้รับผิดชอบเครื่องคอมพิวเตอร์แม่ข่ายทำการอัปเดตทันทีที่สามารถดำเนินการได้

1.5 ตั้งเวลาของเครื่องคอมพิวเตอร์แม่ข่ายโดยใช้มาตรฐานเวลาตามที่ส่วนกลางกำหนดไว้

การตั้งเวลาของเครื่องคอมพิวเตอร์แม่ข่ายให้ตรงกับมาตรฐานเวลาส่วนกลางมีความสำคัญอย่างยิ่งต่อความมั่นคงปลอดภัย ช่วยให้การตรวจสอบบันทึกกิจกรรม (log) มีความถูกต้องแม่นยำ และสามารถระบุเหตุการณ์ด้านความปลอดภัยที่เกิดขึ้นตามลำดับเวลาได้อย่างมีประสิทธิภาพ ซึ่งเป็นสิ่งจำเป็นสำหรับการสืบสวนและแก้ไขปัญหาที่อาจเกิดขึ้นได้ โดยมีแนวปฏิบัติดังนี้

(1) หากเป็นระบบปฏิบัติการ Windows Server ให้ทำการ join domain ของมหาวิทยาลัยพะเยา เพื่อเปิดการตั้งค่าเทียบเวลาของเครื่องคอมพิวเตอร์แม่ข่ายให้โดยอัตโนมัติ

(2) หากเป็นระบบปฏิบัติการอื่นให้ทำการใช้ตั้งค่าเทียบเวลาของเครื่องคอมพิวเตอร์แม่ข่าย โดยกำหนดค่า NTP Server เป็น th.pool.ntp.org

1.6 ห้ามใช้บัญชีผู้ใช้งานที่มหาวิทยาลัยออกสำหรับการปฏิบัติงานทั่วไป เข้าถึงและมีสิทธิ์ administrator ของเครื่องคอมพิวเตอร์แม่ข่าย เพื่อบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่าย

ห้ามใช้บัญชีผู้ใช้งานที่ออกให้โดยมหาวิทยาลัยสำหรับงานทั่วไป ในการเข้าถึงและบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่ายด้วยสิทธิ์ผู้ดูแลระบบ ควรกำหนดบัญชีผู้ดูแลระบบเฉพาะที่มีสิทธิ์ขั้นต่ำที่จำเป็นและใช้สำหรับงานบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่ายเท่านั้น เพื่อลดความเสี่ยงจากการเข้าถึงโดยไม่ได้รับอนุญาต และป้องกันการแพร่กระจายของมัลแวร์ โดยมีแนวปฏิบัติดังนี้

(1) ผู้ดูแลระบบต้องใช้งานบัญชีผู้ใช้งานสำหรับผู้ดูแลระบบที่มหาวิทยาลัยออกให้โดยเฉพาะ (ผู้ดูแลระบบ@up.ac.th) ในการเข้าถึงและการบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่ายเท่านั้น

(2) ผู้ดูแลระบบสามารถสร้างผู้ใช้งานและมอบสิทธิ์การเข้าถึงในระบบปฏิบัติการบนเครื่องคอมพิวเตอร์แม่ข่าย (Local user) ได้

2. ด้าน Secure Web Server

2.1 ใช้ซอฟต์แวร์สำหรับเครื่องคอมพิวเตอร์แม่ข่ายที่ยังมีการสนับสนุนด้านความปลอดภัยอย่างต่อเนื่อง

การเลือกใช้ซอฟต์แวร์เครื่องคอมพิวเตอร์แม่ข่ายที่ยังคงได้รับการสนับสนุนด้านความปลอดภัยอย่างสม่ำเสมอเป็นสิ่งสำคัญอย่างยิ่ง เพื่อให้มั่นใจว่าเครื่องคอมพิวเตอร์แม่ข่ายจะได้รับการอัปเดตช่องโหว่และแพตช์ความปลอดภัยล่าสุดอยู่เสมอ ช่วยลดความเสี่ยงจากการโจมตีที่รู้จักและเสริมสร้างความมั่นคงของระบบโดยรวม โดยให้ใช้งานซอฟต์แวร์เครื่องคอมพิวเตอร์แม่ข่ายบริการเว็บไซต์และเว็บแอปพลิเคชัน (Web Server) รุ่น (version) ที่กำหนดหรือใหม่กว่า ดังนี้

(1) Internet Information Server รุ่น 10.0 บน Windows Server 2019 หรือใหม่กว่า

(2) Apache Web Server รุ่น 2.4 หรือใหม่กว่า

(3) NGINX รุ่น 1.26 หรือใหม่กว่า

2.2 ใช้ภาษาโปรแกรมฝั่งเครื่องคอมพิวเตอร์แม่ข่าย (Server Side Script) ที่มีการสนับสนุนด้านความปลอดภัย

การเลือกใช้ภาษาโปรแกรมฝั่งเครื่องคอมพิวเตอร์แม่ข่ายที่ได้รับการอัปเดตและสนับสนุนด้านความปลอดภัยอย่างต่อเนื่อง เช่น PHP, Python หรือ Node.js เวอร์ชันล่าสุด จะช่วยลดความเสี่ยงจากช่องโหว่ที่มีการเปิดเผยแล้ว และเพิ่มความมั่นคงปลอดภัยโดยรวมของแอปพลิเคชันบนเครื่องคอมพิวเตอร์แม่ข่าย โดยให้ใช้ภาษาโปรแกรมในฝั่งเครื่องคอมพิวเตอร์แม่ข่ายในการพัฒนาเว็บไซต์ที่ปลอดภัย ดังนี้

(1) ASP.NET Framework รุ่น 4.7 หรือใหม่กว่า

(2) ASP.NET Core รุ่น 8.0 หรือใหม่กว่า

(3) PHP รุ่น 8.1 หรือใหม่กว่า

(4) NodeJS รุ่น 22 หรือใหม่กว่า

(5) Python รุ่น 3.9 หรือใหม่กว่า

กรณีใช้ภาษาโปรแกรมอื่นหรือภาษาโปรแกรมรุ่นต่ำกว่าที่กำหนด ให้พิจารณาปรับปรุง (upgrade) โดยด่วน

2.3 ปิดการใช้งาน HTTP (Hyper Text Transfer Protocol)

การปิดใช้งาน HTTP และใช้ HTTPS แทนเป็นสิ่งจำเป็นเพื่อความปลอดภัย เนื่องจาก HTTP ส่งข้อมูลแบบไม่เข้ารหัส ทำให้เสี่ยงต่อการดักจับข้อมูล การเปลี่ยนไปใช้ HTTPS จะช่วยให้การสื่อสารทั้งหมดถูกเข้ารหัส เพิ่มความปลอดภัยให้กับข้อมูลสำคัญบนเครื่องคอมพิวเตอร์แม่ข่าย โดยมีแนวปฏิบัติดังนี้

(1) ตั้งค่าให้เครื่องคอมพิวเตอร์แม่ข่ายปฏิเสธการเชื่อมต่อทั้งหมดผ่านพอร์ต 80 จากระบบเครือข่ายอินเทอร์เน็ต

(2) ตั้งค่าให้เครื่องคอมพิวเตอร์แม่ข่ายใช้บริการ HTTPS ผ่านพอร์ต 443 เท่านั้น

(3) หากส่วนงานหรือหน่วยงานยังไม่มีการจัดหาใบรับรองความปลอดภัยสำหรับบริการ HTTPS ให้ใช้บริการ Free Public Digital Certificate เช่น Let’s Encrypt

(4) ทดสอบการเข้าถึงเว็บไซต์ผ่านพอร์ต 80 และ 443 จากระบบเครือข่ายภายนอกมหาวิทยาลัย เพื่อให้มั่นใจว่าเป็นไปตามแนวปฏิบัติ โดยใช้เครื่องมือภายนอก เช่น https://developer.mozilla.org/en-US/observatory

2.4 เปิดการใช้งาน HSTS (HTTP Strict Transport Security)

การเปิดใช้งาน HSTS เป็นการบังคับให้เบราว์เซอร์เชื่อมต่อกับเครื่องคอมพิวเตอร์แม่ข่ายด้วยโปรโตคอล HTTPS เสมอ เพื่อป้องกันการโจมตีแบบ Downgrade Attack และการดักจับข้อมูล (Man-in-the-Middle) ทำให้มั่นใจได้ว่าการสื่อสารทั้งหมดมีความปลอดภัยและเข้ารหัสอย่างสมบูรณ์ โดยมีแนวปฏิบัติดังนี้

(1) ตั้งค่า HTTP Header ของเครื่องคอมพิวเตอร์แม่ข่ายที่ใช้งาน โดยให้เพิ่มค่า Strict-Transport-Security เป็นการตั้งค่าเริ่มต้น (default setting) โดยกำหนดค่า max-age = ค่าสูงสุดตามข้อกำหนดของเครื่องคอมพิวเตอร์แม่ข่ายที่ใช้ (เช่น 31536000 เท่ากับ 1 ปี) รวมทั้งกำหนดค่า includeSubDomains และค่า preload ร่วมไปด้วย

(2) ทดสอบการเข้าถึงเว็บไซต์ผ่านพอร์ต 80 และ 443 จากระบบเครือข่ายภายนอกมหาวิทยาลัย เพื่อให้มั่นใจว่าเป็นไปตามแนวปฏิบัติ โดยใช้เครื่องมือภายนอก เช่น https://developer.mozilla.org/en-US/observatory

2.5 ปิดการใช้งาน Algorithm การเข้ารหัส HTTPS ที่ไม่ปลอดภัย

การปิดใช้งาน Algorithm การเข้ารหัส HTTPS ที่ล้าสมัยหรือไม่ปลอดภัย จะช่วยป้องกันการดักฟังข้อมูลและการโจมตีที่อาศัยช่องโหว่ของ Algorithm เหล่านั้น ทำให้การสื่อสารผ่าน HTTPS มีความปลอดภัยและเป็นส่วนตัวมากยิ่งขึ้น โดยมีแนวปฏิบัติดังนี้

(1) ปิดการเข้ารหัสมาตรฐาน TLS 1.0, TLS 1.1, SSL 2.0, SSL 3.0

(2) ใช้การเข้ารหัสมาตรฐาน TLS 1.2 หรือดีกว่า

(3) ใช้การเข้ารหัสใบรับรองความปลอดภัย แบบ SHA-256 หรือดีกว่า

(4) ทดสอบการปิดและการใช้ algorithm ที่ปลอดภัยโดยใช้บริการตรวจสอบ เช่น https://www.ssllabs.com/ssltest

2.6 กำหนดสิทธิ์การเข้าถึงไฟล์ (file) และโฟลเดอร์ (folder) อย่างปลอดภัยตามสิทธิ์การใช้งาน

การกำหนดสิทธิ์การเข้าถึงไฟล์และโฟลเดอร์อย่างรัดกุมเป็นสิ่งสำคัญ เพื่อให้แน่ใจว่าผู้ใช้งานมีสิทธิ์เข้าถึงเฉพาะข้อมูลที่จำเป็นต่อการทำงานเท่านั้น ซึ่งจะช่วยป้องกันการเข้าถึงโดยไม่ได้รับอนุญาตและลดความเสี่ยงด้านความปลอดภัยของข้อมูลบนเครื่องคอมพิวเตอร์แม่ข่าย โดยมีแนวปฏิบัติดังนี้

(1) อนุญาตให้ผู้ใช้งานทั่วไป (everyone) หรือ ผู้ใช้งานที่เป็น process ของ web server สามารถเข้าถึงไฟล์ webpage ภายในโฟลเดอร์ เช่น wwwroot หรือ htdocs หรือ var/www แบบอ่านอย่างเดียวเท่านั้น

(2) ห้ามไม่ให้ผู้ใช้งานทั่วไป (everyone) หรือ ผู้ใช้งานที่เป็น process ของ web server สามารถเข้าถึงไฟล์หรือโฟลเดอร์ อื่นๆ นอกเหนือจาก (1)

(3) ปิดการแสดงไฟล์และโฟลเดอร์โดยตรง (directory browsing) ผ่าน web browser หรือผ่าน HTTP

2.7 ให้สิทธิ์ต่ำที่สุด (least privilege) สำหรับ process ของ web server

การตั้งค่าให้กระบวนการของเครื่องคอมพิวเตอร์แม่ข่ายมีสิทธิ์ขั้นต่ำที่สุดที่จำเป็นต่อการทำงาน จะช่วยลดความเสียหายที่อาจเกิดขึ้น หากระบบถูกบุกรุก สิทธิ์ที่จำกัดนี้จะป้องกันผู้ไม่ประสงค์ดีจากการเข้าถึงหรือแก้ไขข้อมูลสำคัญเกินความจำเป็น โดยมีแนวปฏิบัติดังนี้

(1) IIS ควรใช้บัญชีผู้ใช้งานที่มีสิทธิ์อ่านอย่างเดียวในการ Run Process ไม่ควรใช้บัญชีผู้ใช้งานที่เป็น administrator หรือบัญชีผู้ใช้งานส่วนตัวที่มหาวิทยาลัยกำหนดให้ ควรใช้ local account ที่สร้างภายในเครื่อง และในส่วนของ Linux ไม่ควรให้สิทธิระดับ root ในการ run background service ต่าง ๆ

(2) ไม่ใช้บัญชีผู้ใช้งานส่วนตัวที่ทางมหาวิทยาลัยออกให้ หรือ Administrator หรือ root หรือ System ที่เป็นบัญชีที่มีสิทธิ์ระดับสูงสุด ในการ Run Process ของ Web Server

(3) ใช้ Local Account และกำหนดสิทธิ์ “Read-Only” ในการ Run Process ของ Web Server

3. ด้าน Secure Database Server

3.1 อนุญาตเฉพาะเว็บไซต์หรือแอปพลิเคชันที่พัฒนาให้สามารถเข้าถึง database server ได้เท่านั้น

เพื่อความปลอดภัยสูงสุด จำกัดการเข้าถึง database server เฉพาะเว็บไซต์หรือแอปพลิเคชันที่พัฒนาขึ้นและได้รับอนุญาตเท่านั้น การทำเช่นนี้จะช่วยป้องกันการเข้าถึงจากแหล่งที่ไม่ได้รับอนุญาต ลดความเสี่ยงของการโจมตีและการรั่วไหลของข้อมูลได้อย่างมีนัยสำคัญ โดยมีแนวปฏิบัติดังนี้

(1) ปิดการเข้าถึง Database Server จาก Internet โดยตรง

(2) ใช้ Firewall ของเครื่องหรือระบบฐานข้อมูล ในการกำหนด IP Address ของ Web Server หรือ Web Application และเครื่องคอมพิวเตอร์ของผู้ดูแลระบบ หรือระบุ IP Address ต้นทางที่สามารถเข้าถึงเครื่องคอมพิวเตอร์แม่ข่ายได้เท่านั้น

3.2 ปิดการใช้งานบริการ background service ที่ไม่จำเป็นใน database server

การปิดบริการพื้นหลังที่ไม่จำเป็นบนเครื่องคอมพิวเตอร์แม่ข่ายฐานข้อมูลจะช่วยลดความเสี่ยงด้านความปลอดภัย โดยจำกัดช่องโหว่ที่อาจถูกโจมตีได้ การดำเนินการนี้ยังช่วยเพิ่มประสิทธิภาพของเครื่องคอมพิวเตอร์แม่ข่ายด้วยการลดการใช้ทรัพยากรที่ไม่จำเป็น ตัวอย่างบริการที่ไม่จำเป็นและควรปิด ได้แก่

(1) Web server เช่น IIS, NGINX และ Apache

(2) FTP server

(3) Webdav

(4) PhpMyAdmin

(5) Content Management System เช่น Joomla

3.3 ตรวจสอบบัญชีผู้ใช้งาน (username และ password) อย่างน้อยทุก 6 เดือนและลบบัญชีที่ไม่ใช้งานแล้ว

เพื่อรักษาความปลอดภัยของระบบ ควรตรวจสอบบัญชีผู้ใช้งานทั้งหมดอย่างน้อยทุก 6 เดือน โดยลบบัญชีที่ไม่มีการใช้งาน และเปลี่ยนแปลงรหัสผ่านสำหรับบัญชีที่ยังใช้งานอยู่ เพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาตและลดความเสี่ยงจากการโจมตี โดยมีแนวปฏิบัติดังนี้

(1) รหัสผ่าน (password) ควรกำหนดรหัสผ่านที่มีความแข็งแรง โดยมีความยาวขั้นต่ำ 12 ตัวอักษร และประกอบด้วยตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก ตัวเลขและสัญลักษณ์พิเศษ

(2) เปลี่ยนรหัสผ่าน เมื่อคาดว่ารหัสผ่านอาจมีการรั่วไหล

(3) ให้ตรวจสอบบัญชีผู้ใช้งาน database server และลบบัญชีผู้ใช้งานที่ไม่มีการใช้งานหรือไม่มีความจำเป็นในการเข้าถึง

3.4 ปิดบัญชีผู้ใช้งานฐานข้อมูลที่เป็นค่าเริ่มต้น

การปิดหรือเปลี่ยนรหัสผ่านบัญชีผู้ใช้งานฐานข้อมูลที่เป็นค่าเริ่มต้น (default accounts) จะช่วยป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต เนื่องจากผู้ไม่ประสงค์ดีมักจะทราบข้อมูลบัญชีที่เป็นค่าเริ่มต้นเหล่านี้ ทำให้ระบบมีความปลอดภัยมากยิ่งขึ้นจากการโจมตีใช้ข้อมูลประจำตัวที่เป็นค่าเริ่มต้น โดยมีแนวปฏิบัติดังนี้

(1) ทำการปิดการใช้งาน (disable) หรือลบบัญชีผู้ใช้งานที่ใช้ชื่อเป็นค่าเริ่มต้น เช่น sa, root หรือ administrator ออกจากระบบ database server

(2) ทำการเปลี่ยนรหัสผ่านที่เป็นค่าเริ่มต้น (default password) สำหรับทุกบริการในเครื่องคอมพิวเตอร์แม่ข่าย

3.5 สำรองข้อมูล database อัตโนมัติหรือทำด้วยตนเอง นอกเหนือจากการให้ส่วนกลางสำรองข้อมูล database server

ควรกำหนดให้มีการสำรองข้อมูล Database เป็นประจำ ไม่ว่าจะแบบอัตโนมัติหรือด้วยตนเอง นอกเหนือจากระบบสำรองข้อมูลส่วนกลาง การทำเช่นนี้จะช่วยให้มั่นใจว่ามีข้อมูลสำรองล่าสุดพร้อมใช้งานอยู่เสมอ เพื่อให้สามารถกู้คืนระบบได้อย่างรวดเร็วในกรณีที่เกิดความเสียหายหรือเกิดเหตุการณ์ที่ไม่อาจคาดการณ์ล่วงหน้าได้ โดยมีแนวปฏิบัติดังนี้

(1) สำรองข้อมูลไฟล์ฐานข้อมูลของส่วนงานหรือหน่วยงานด้วยตนเอง ตามความเหมาะสมและความถี่ที่สอดคล้องกับความต้องการ เพื่อให้มั่นใจว่าหากต้องมีการกู้คืนจะได้ข้อมูลที่ล่าสุดเสมอ

(2) ทดสอบการนำข้อมูลที่สำรองไว้มากู้คืนในเครื่องคอมพิวเตอร์สำหรับการทดสอบ หรือกู้คืนไปไว้ยัง database สำหรับการทดสอบ เพื่อให้มั่นใจว่าข้อมูลที่ถูกสำรองไว้สามารถกู้คืนและนำไปใช้งานได้

3.6 ห้ามใช้บัญชีผู้ใช้งานที่มหาวิทยาลัยออกให้สำหรับการปฏิบัติงานทั่วไป ในการเชื่อมต่อกับ Database Server หรือใช้เป็นบัญชีผู้ดูแลระบบ database server

เพื่อความปลอดภัยสูงสุด ห้ามใช้บัญชีผู้ใช้งานที่ออกโดยมหาวิทยาลัยสำหรับปฏิบัติงานทั่วไปในการเชื่อมต่อหรือเป็นผู้ดูแลระบบ Database Server โดยเด็ดขาด ให้ใช้บัญชีเฉพาะที่สร้างขึ้นสำหรับงาน Database โดยมีสิทธิ์เข้าถึงเท่าที่จำเป็นเท่านั้น เพื่อลดความเสี่ยงจากการเข้าถึงโดยไม่ได้รับอนุญาต และจำกัดผลกระทบหากบัญชีทั่วไปถูกบุกรุก โดยมีแนวปฏิบัติดังนี้

(1) สร้างบัญชีผู้ใช้งานสำหรับเครื่องคอมพิวเตอร์แม่ข่าย (local account) เพื่อใช้ในการบริหารจัดการ Database Server โดยเฉพาะ

(2) ห้ามใช้บัญชีผู้ใช้งานที่สร้างขึ้นกับบริการอื่น ๆ ในเครื่องคอมพิวเตอร์แม่ข่าย

3.7 กำหนดสิทธิ์การเข้าถึงแก้ไขเปลี่ยนข้อมูลใน database server ให้เหมาะสม

การกำหนดสิทธิ์การเข้าถึงข้อมูลใน Database Server อย่างเหมาะสมเป็นสิ่งสำคัญยิ่ง โดยให้สิทธิ์ผู้ใช้งานและแอปพลิเคชันเท่าที่จำเป็นต่อการปฏิบัติงาน (Principle of Least Privilege) เพื่อป้องกันการเข้าถึง แก้ไข หรือเปลี่ยนแปลงข้อมูลโดยไม่ได้รับอนุญาต ซึ่งจะช่วยลดความเสี่ยงจากการรั่วไหลหรือความเสียหายของข้อมูลได้ โดยมีแนวปฏิบัติดังนี้

(1) สำหรับผู้ใช้งานหรือระบบที่ต้องการอ่านข้อมูลอย่างเดียว ให้กำหนดสิทธิ์แบบ read only และระบุ table หรือ view ที่อนุญาตให้อ่านได้เท่านั้น ไม่ควรให้สิทธิ์เข้าถึง database ได้ทั้งหมด

(2) ทำการตรวจสอบสิทธิ์ที่มอบให้ และทำการยกเลิกสิทธิ์ หรือลบ user ออกจากระบบ กรณีไม่จำเป็นต้องใช้งานอีกแล้ว

4. ด้าน Secure Application Development

4.1 ปรับปรุงซอฟต์แวร์สำหรับการพัฒนาเว็บไซต์อยู่เสมอ

การอัปเดตซอฟต์แวร์และเฟรมเวิร์กที่ใช้ในการพัฒนาเว็บไซต์ให้เป็นปัจจุบันอยู่เสมอจะช่วยลดความเสี่ยงจากช่องโหว่ที่มีการเปิดเผยแล้ว ซึ่งผู้ไม่ประสงค์ดีมักใช้โจมตี การดำเนินการนี้เป็นการเสริมสร้างความปลอดภัยโดยรวมของเครื่องคอมพิวเตอร์แม่ข่าย โดยมีแนวปฏิบัติดังนี้

(1) ทำการรายการซอฟต์แวร์หรือเฟรมเวิร์กที่ใช้ในการพัฒนาระบบสารสนเทศ ที่ระบุชื่อ เวอร์ชัน วันที่ปรับปรุงล่าสุด

(2) ตรวจสอบเวอร์ชันตามรายการซอฟต์แวร์หรือเฟรมเวิร์กกับผู้ผลิตหรือผู้พัฒนาซอฟต์แวร์ดังกล่าว ว่าได้รับการปรับปรุงเป็นรุ่นล่าสุด หรือเป็นรุ่นที่ปลอดภัย และยังมีการสนับสนุนด้านความปลอดภัย

(3) ทำการอัปเดตเพื่อป้องกันช่องโหว่ด้านความปลอดภัยให้กับซอฟต์แวร์หรือเฟรมเวิร์กที่ใช้ในการพัฒนาระบบสารสนเทศตลอดเวลา โดยเฉพาะเมื่อได้รับการแจ้งเตือนจากศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์ หรือจากแหล่งที่มีความน่าเชื่อถือ

4.2 ปิดการแสดงข้อความผิดพลาดใน Web Browser

การปิดการแสดงข้อความผิดพลาดใน Web Browser เป็นสิ่งจำเป็นเพื่อป้องกันไม่ให้ผู้โจมตีได้รับข้อมูลสำคัญเกี่ยวกับโครงสร้างภายในของระบบ ซึ่งการเปิดเผยข้อผิดพลาดโดยไม่ตั้งใจอาจนำไปสู่การเปิดเผยข้อมูลช่องโหว่และทำให้เครื่องคอมพิวเตอร์แม่ข่ายตกอยู่ในความเสี่ยง โดยมีแนวปฏิบัติดังนี้

(1) ตั้งค่าการปิดการแสดงข้อมูลผิดพลาดตามวิธีการที่กำหนดไว้ของเฟรมเวิร์กหรือซอฟต์แวร์ที่ใช้ในการพัฒนาระบบสารสนเทศและเว็บไซต์

(2) ทำการตรวจสอบ โดยการสร้างชุดคำสั่งที่ทำให้เกิดข้อผิดพลาด และทำการเข้าถึงชุดคำสั่งดังกล่าวผ่านทาง web browser หรือคำสั่งเช่น curl เพื่อตรวจสอบการทำงานของการปิดการแสดงข้อมูลผิดพลาด

4.3 ปิดการทำงาน debugging mode

การปิดใช้งาน debugging mode ในสภาพแวดล้อมจริงเป็นสิ่งสำคัญ เนื่องจากโหมดนี้สามารถเปิดเผยข้อมูลที่มีความละเอียดอ่อนเกี่ยวกับระบบและการกำหนดค่า อาจทำให้เกิดความเสี่ยงต่อการรั่วไหลของข้อมูลและการโจมตี การดำเนินการดังกล่าวจึงมีส่วนช่วยในการลดความเสี่ยงและป้องกันการโจมตีที่อาจจะเกิดขึ้นได้ โดยมีแนวปฏิบัติดังนี้

(1) ปิดการใช้งานสภาพแวดล้อมแบบ debug เมื่อทำการติดตั้ง (deploy) เว็บไซต์และเว็บแอปพลิเคชันในเครื่องคอมพิวเตอร์แม่ข่ายเพื่อใช้งานจริง (production stage)

(2) ตั้งค่าการเขียนไฟล์บันทึกข้อผิดพลาด (error log) เมื่อเกิดข้อผิดพลาดในการทำงานของเว็บไซต์และเว็บแอปพลิเคชัน แทนการใช้งาน debug

4.4 ตั้งค่าการใช้ Session Cookies ให้ปลอดภัย

การตั้งค่า Session Cookies ให้ปลอดภัยเป็นสิ่งจำเป็นเพื่อป้องกันการโจรกรรม Session โดยกำหนดให้ใช้ Secure และ HttpOnly flag ซึ่งช่วยให้มั่นใจได้ว่า cookies จะถูกส่งผ่านช่องทางที่เข้ารหัสเท่านั้นและไม่สามารถเข้าถึงได้ผ่านสคริปต์ฝั่งไคลเอนต์ ทำให้ลดความเสี่ยงจากการโจมตีแบบ Cross-Site Scripting (XSS) ได้ โดยมีแนวปฏิบัติดังนี้

(1) กรณีที่ใช้ Session Cookies ควรกำหนดค่า SameSite Cookie ให้เป็น strict หรือ Lax เพื่อป้องกันการโจมตีแบบ CSRF

(2) กำหนดค่า secure flag เป็น secure เพื่อให้ web browser ใช้งาน Session Cookies กับการรับส่งข้อมูลที่ปลอดภัย HTTPS เท่านั้น

(3) กำหนดค่า HttpOnly flag เพื่อป้องการการเข้าถึง cookies ด้วย script ต่าง ๆ

4.5 กำหนดระยะเวลาการหมดอายุของ user session และ JWT (Json web token) ให้เหมาะสม

เมื่อมีการใช้งาน user session ไม่ว่าจะเป็นแบบ cookies, query หรือ JWT ควรมีการระบุเวลาหมดอายุของ session หรือ token เหล่านั้น ไม่ควรให้สามารถใช้งานได้ตลอดหรือใช้งานได้เป็นระยะเวลาหลายวัน เพื่อป้องกันการขโมย session ไปใช้งานโดยผู้ไม่ประสงค์ดี โดยมีแนวปฏิบัติดังนี้

(1) ควรมีการตั้งค่าระยะเวลาที่ user session หมดอายุ (idle timeout) เมื่อไม่มีการติดต่อกับเครื่องคอมพิวเตอร์แม่ข่ายเป็นระยะเวลานาน สำหรับระบบที่ต้องการการเข้าสู่ระบบเพื่อใช้งาน

(2) มีการตั้งค่าระยะเวลาหมดอายุสูงสุดของ user session (absolute timeout) โดยไม่คำนึงถึงกิจกรรมของผู้ใช้งาน เพื่อป้องกันการโจมตีแบบ session hijack เช่น หากระบบมีการใช้งานตลอดช่วงเวลาปฏิบัติงานปกติ ควรกำหนดระยะเวลาหมดอายุไว้ที่ 7-8 ชั่วโมง

(3) ควรมีการตั้งระยะเวลาการหมดอายุของ access token สำหรับแอปพลิเคชันที่ต้องการความปลอดภัยสูง ควรตั้งระยะเวลาหมดอายุไว้ที่ 5-15 นาที สำหรับแอปพลิเคชันทั่วไปควรตั้งระยะเวลาหมดอายุไว้ที่ 1-24 ชั่วโมง

(4) ควรมีการตั้งระยะเวลาหมดอายุของ refresh token ไว้ที่ 7-30 วัน และควรใช้ refresh token แบบใช้ครั้งเดียว (one-time-use)

4.6 ตรวจสอบข้อมูลนำเข้า (input validation) จากผู้ใช้งานก่อนประมวลผลเสมอ

การตรวจสอบความถูกต้องของข้อมูลที่รับจากผู้ใช้งานก่อนการประมวลผลเป็นสิ่งจำเป็น เพื่อป้องกันช่องโหว่จากการโจมตี เช่น SQL Injection หรือ Cross-Site Scripting (XSS) ซึ่งช่วยให้มั่นใจได้ว่ามีเพียงข้อมูลที่ถูกต้องและปลอดภัยเท่านั้นที่เข้าสู่ระบบสารสนเทศ โดยมีแนวปฏิบัติดังนี้

(1) เขียนคำสั่งให้มีการตรวจสอบข้อมูลนำเข้าในฝั่งเครื่องแม่ข่าย (server-side validation)

(2) เขียนคำสั่งให้มีการกำหนดอักขระ (character) หรือข้อมูลใดบ้างที่สามารถส่งจากไคลแอนต์มายังเครื่องคอมพิวเตอร์แม่ข่ายได้ (whitelist) และหลีกเลี่ยงการกรองข้อมูลโดยใช้รายการคำหรืออักขระ (blacklist) เช่น การกรอกเลขบัตรประจำตัวประชาชนรับเฉพาะตัวเลข 0-9 จำนวน 13 อักขระ เป็นต้น

(3) เขียนคำสั่งตรวจสอบข้อมูลโดยใช้ Regular Expression (Regex) เพื่อให้ครอบคลุมรายการข้อมูลที่ต้องการ

(4) ป้องกัน SQL Injection ด้วยการใช้ Prepared Statements หรือ Parameterized Queries จะแยกคำสั่ง SQL ออกจากข้อมูลอย่างชัดเจน หรือใช้ไลบรารี (library) สำหรับจัดการข้อมูลในแบบเชิงวัตถุหรือ ORM (Object Relation Mapping)

(5) การตรวจสอบข้อมูลนำเข้าที่จะมาทำการประมวลผลในเว็บแอปพลิเคชันต้องทำการตรวจสอบทั้งหมด ทั้งจาก form, query string, header, cookie และการอัปโหลดไฟล์ (ตรวจสอบขนาดไฟล์ ประเภทไฟล์ นามสกุล content type เป็นต้น)

4.7 ป้องกันการโจมตีแบบ CSRF (Cross-site Request Forgery) โดยการใช้งาน CORS (Cross-Origin Resource Sharing) หรือ Anti Request Forgery Token

การป้องกัน CSRF (Cross-Site Request Forgery) สามารถทำได้โดยการใช้ CORS (Cross-Origin Resource Sharing) เพื่อจำกัดโดเมนที่สามารถส่งคำขอได้ หรือการใช้ Anti-Forgery Token ที่ไม่ซ้ำกันในทุกคำขอที่มีการเปลี่ยนแปลงสถานะ ช่วยให้มั่นใจได้ว่าคำขอมาจากผู้ใช้จริง ไม่ใช่การปลอมแปลง โดยมีแนวปฏิบัติดังนี้

(1) กำหนดค่า CORS ที่ถูกต้องช่วยเสริมความปลอดภัยโดยรวมของระบบ โดยการจำกัดว่าสคริปต์จาก Origin อื่นสามารถอ่าน (Response) หรือสร้างคำขอ (Request) มายังเครื่องคอมพิวเตอร์แม่ข่ายได้

(2) สำหรับทุก session ของผู้ใช้ ให้เครื่องคอมพิวเตอร์แม่ข่ายสร้าง Token ที่ไม่ซ้ำกัน, สุ่ม และคาดเดาไม่ได้ (เช่น GUID หรือสตริงสุ่มที่มีความยาวมากพอ) และเชื่อมโยง Token นี้เข้ากับ session ของผู้ใช้ และเขียนคำสั่งที่ client ให้ทำการส่ง Token ดังกล่าวมาทุกครั้งร่วมกันกับข้อมูลที่ต้องทำการประมวลผลที่เครื่องคอมพิวเตอร์แม่ข่าย เพื่อให้มั่นใจว่ามาจากแหล่งที่ถูกต้อง

4.8 ระบุ X-Content-Type-Options nosniff เพื่อให้ Web Browser ไม่พยายามคาดเดาประเภทไฟล์เอง

การกำหนด HTTP Header X-Content-Type-Options: nosniff จะช่วยป้องกันเบราว์เซอร์ไม่ให้พยายาม “คาดเดา” ชนิดของไฟล์ (MIME type) ที่ไม่ตรงกับที่เซิร์ฟเวอร์ระบุ ซึ่งจะช่วยลดความเสี่ยงจากการโจมตีประเภท MIME-sniffing หรือ XSS ที่เกิดจากการเข้าใจผิดประเภทของไฟล์ โดยมีแนวปฏิบัติดังนี้

(1) ทำการระบุค่า Header X-Content-Type-Options: nosniff ให้กับ ทุก HTTP Response ที่เครื่องคอมพิวเตอร์แม่ข่ายส่งกลับไปยัง client

(2) ทำการระบุค่า Header Content-Type ให้กับ ทุก HTTP Response ที่ส่งกลับไปยัง client ให้ถูกต้องตามประเภทข้อมูล เช่น หากเป็น JSON ระบุเป็น application/json เป็นต้น

4.9 ใช้งาน Sub Resource Integrity (SRI) เมื่อมีการใช้งาน script จากผู้ให้บริการ Content Delivery Network (CDN)

การใช้ Sub Resource Integrity (SRI) เมื่อดึงสคริปต์หรือสไตล์ชีทจาก CDN ช่วยให้เบราว์เซอร์ตรวจสอบค่า Hash ของไฟล์ที่ดาวน์โหลดว่าตรงกับค่าที่ระบุไว้หรือไม่ เพื่อยืนยันว่าไฟล์นั้นไม่มีการเปลี่ยนแปลงหรือถูกดัดแปลงโดยผู้ไม่หวังดี ทำให้เพิ่มความมั่นคงปลอดภัยจากการโจมตีประเภท Man-in-the-Middle ได้ โดยมีแนวปฏิบัติดังนี้

(1) ทุกครั้งที่มีการใช้งานไฟล์ JavaScript หรือ CSS ที่จัดเก็บไว้ในผู้ให้บริการ CDN ให้เพิ่ม integrity attribute เข้าไปในแท็ก <script> หรือ <link> เสมอ กรณีผู้ให้บริการไม่ได้แจ้งค่า hash สำหรับใช้ในการระบุ integrity ให้ใช้เครื่องมือสร้าง hash เพื่อระบุค่าด้วยตัวเอง เช่น https://www.srihash.org/

(2) ระบุค่า crossorigin=“anonymous” attribute ให้กับ <script> และ <link> tags เสมอ เพื่อให้เบราว์เซอร์ทำการร้องขอแบบ CORS “anonymous” ซึ่งจำเป็นสำหรับการทำงานของ SRI และป้องกันไม่ให้ Credential (เช่น Cookies) ถูกส่งไปพร้อมกับคำขอที่ไม่เกี่ยวข้อง

4.10 กำหนดนโยบายการแสดงแหล่งที่มา (Referrer Policy) เมื่อผู้ใช้งาน Click link เพื่อไปเว็บไซต์อื่น

การกำหนด Referrer Policy ช่วยควบคุมว่าข้อมูล URL ของหน้าปัจจุบันจะถูกส่งไปยังเว็บไซต์ปลายทางเมื่อผู้ใช้งานคลิกลิงก์หรือไม่ ซึ่งช่วยปกป้องความเป็นส่วนตัวของผู้ใช้งานและลดการรั่วไหลของข้อมูลที่ละเอียดอ่อนผ่าน HTTP Referrer Header ได้อย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) ทำการกำหนดค่า Referrer-Policy: no-referrer ให้กับ HTTP header ทุก response ที่ส่งกลับไปยัง client

(2) กรณีต้องการกำหนด Referrer Policy เฉพาะบ้างหน้า (page) ให้ใช้ meta name=“referrer” content=“no-referrer” ในส่วน header ของเอกสาร HTML

(3) กรณีต้องการกำหนด Referrer Policy เฉพาะใน <a> tag ให้กำหนดค่า rel=“no-referrer”

4.11 ระบุค่า Content Security Policy (CSP) ใน HTTP Header เพื่อจำกัดแหล่งที่มาของ script ที่ไม่ปลอดภัย

การกำหนด HTTP Header Content Security Policy (CSP) ช่วยให้สามารถระบุแหล่งที่มาที่เชื่อถือได้ของเนื้อหา เช่น สคริปต์ สไตล์ชีท และรูปภาพ เพื่อป้องกันการโจมตีแบบ Cross-Site Scripting (XSS) และการแทรกโค้ดอันตรายอื่น ๆ เข้ามาในหน้าเว็บอย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) ทำการกำหนดค่า Content-Security-Policy: default-src 'self' ให้กับ HTTP header ทุก response ที่ส่งกลับไปยัง client หมายความว่า ทรัพยากรทั้งหมด (สคริปต์, สไตล์ชีท, รูปภาพ ฯลฯ) จะถูกอนุญาตให้โหลดได้จาก Origin เดียวกันกับเว็บไซต์เท่านั้น

(2) ทำการตรวจสอบการใช้งาน Content Security Policy ด้วยเครื่องมือตรวจสอบ เช่น https://csp-evaluator.withgoogle.com/

4.12 ระบุค่า X-Frame-Options เป็น DENY ใน header ของการตอบกลับ HTTP

การตั้งค่า HTTP Header X-Frame-Options: DENY ในการตอบกลับ HTTP เป็นมาตรการป้องกันเบราว์เซอร์ไม่ให้แสดงหน้าเว็บภายใน <iframe>, <frame> หรือ <object> บนเว็บไซต์อื่น ช่วยลดความเสี่ยงจากการโจมตีแบบ Clickjacking ได้อย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) ทำการกำหนดค่า X-Frame-Options: DENY ให้กับ HTTP header ทุก response ที่ส่งกลับไปยัง client หมายความว่า web browser จะปฏิเสธไม่ให้หน้าเว็บถูกฝังในเฟรมใด ๆ ไม่ว่าจะเป็น Origin เดียวกันหรือข้าม Origin ก็ตาม เพื่อป้องกันการโจมตี Clickjacking ผ่านเฟรม

(2) กรณีมีความจำเป็นต้องใช้ iFrame อาจกำหนดค่า X-Frame-Options: SAMEORIGIN เพื่อป้องกันการฝัง script จากภายนอก

4.13 สมัครใช้บริการ Google Search Console เพื่อแก้ไขปัญหาฝากลิงค์เว็บพนัน

การสมัครและใช้งาน Google Search Console ช่วยให้สามารถตรวจสอบประสิทธิภาพของเว็บไซต์ในผลการค้นหาของ Google รวมถึงตรวจจับปัญหาด้านความปลอดภัย เช่น การถูกฝังลิงก์เว็บพนัน (Spam/Hacked Content) ทำให้สามารถระบุแหล่งที่มาและดำเนินการแก้ไขเพื่อลบลิงก์ที่ไม่พึงประสงค์ออกและกู้คืนความน่าเชื่อถือของเว็บไซต์ได้อย่างรวดเร็ว โดยมีแนวปฏิบัติดังนี้

(1) สมัครใช้งาน Google Search Console (หากยังไม่มี) และยืนยันสิทธิ์การเป็นเจ้าของเว็บไซต์

(2) เข้าสู่ระบบ Search Console ให้ไปที่เมนู “ความปลอดภัยและการดำเนินการด้วยตนเอง” (Security & Manual Actions) -> “ปัญหาด้านความปลอดภัย” (Security issues) เพื่อตรวจสอบว่าเว็บไซต์ถูกแฮกหรือมีเนื้อหาสแปมที่ถูกแทรกเข้ามาหรือไม่

(3) หาก URL ที่มีสแปมถูก Google Index เรียบร้อยแล้ว ให้ใช้เครื่องมือ “การนำ URL ออก” (Removals) ใน Search Console เพื่อร้องขอให้ Google ลบ URL เหล่านั้นออกจากผลการค้นหาชั่วคราว

4.14 ตรวจสอบการนำเข้าข้อมูลใน HTML form โดยใช้ Captcha หรือเทคนิคอื่น ๆ สำหรับแบบฟอร์มที่เป็นสาธารณะ

การใช้ Captcha หรือเทคนิคการตรวจสอบอื่น ๆ กับแบบฟอร์มสาธารณะ (Public HTML Form) เป็นสิ่งจำเป็นเพื่อยืนยันว่าผู้ที่ส่งข้อมูลเป็นมนุษย์ ไม่ใช่บอทอัตโนมัติ (bot) ซึ่งช่วยป้องกันการโจมตีแบบสแปม, การสมัครสมาชิกปลอม หรือการยิงข้อมูลจำนวนมากเข้าสู่ระบบได้อย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) ใช้บริการ CAPTCHA เช่น reCAPTCHA v3 ของ Google

(2) ใช้เทคนิคเพิ่มเติม เช่น การจำกัดอัตราการส่ง (Rate Limiting) การตรวจสอบเวลาที่ใช้กรอกฟอร์ม (Time-based Honeypot) หรือวิเคราะห์รูปแบบการส่งข้อมูล เช่น จำนวนช่องที่กรอก, ความยาวของข้อความ, พฤติกรรมการคลิก หากพบรูปแบบที่ผิดปกติ (เช่น กรอกฟิลด์ทั้งหมดสมบูรณ์แบบในเวลาอันสั้น) อาจบล็อกหรือตรวจสอบเพิ่มเติม

4.15 ควบคุมการทำดัชนีการค้นหาของบริการค้นหา (search engine)

การควบคุมการทำดัชนีของ Search Engine ช่วยให้คุณกำหนดได้ว่าส่วนใดของเว็บไซต์ที่ควรปรากฏในผลการค้นหา เพื่อป้องกันข้อมูลที่ละเอียดอ่อนหรือหน้าเว็บที่ไม่ควรเปิดเผยต่อสาธารณะจากการถูกเข้าถึง ช่วยเสริมความมั่นคงปลอดภัยและความเป็นส่วนตัวของระบบโดยรวม โดยมีแนวปฏิบัติดังนี้

(1) ใช้ไฟล์ robots.txt เพื่อบอก Search Engine Crawlers ว่าส่วนใดของเว็บไซต์ที่ไม่ควรทำการ Crawl หรือ Index ตัวอย่างส่วนที่ไม่ควรทำ Index เช่น หน้าเข้าสู่ระบบและหน้าบัญชีผู้ใช้ ตะกร้าสินค้าและหน้าชำระเงิน ไฟล์และไดเรกทอรีใช้เฉพาะภายใน และหน้าทดสอบหรือหน้าสำหรับการพัฒนา เป็นต้น

(2) ใช้ Meta Tag noindex หรือ HTTP Header X-Robots-Tag: noindex ในหน้าที่ไม่ต้องการให้ search engine ทำ Index

4.16 ไม่เก็บ Source Code ของ Web Application ไว้ใน Production Web Server

การไม่เก็บ Source Code ของ Web Application บน Production Web Server โดยตรง จะช่วยลดความเสี่ยงจากการถูกเปิดเผยโค้ดเมื่อเครื่องคอมพิวเตอร์แม่ข่ายถูกโจมตี ทำให้ผู้ไม่หวังดีไม่สามารถเข้าถึงข้อมูลเชิงลึกของระบบหรือค้นหาช่องโหว่เพิ่มเติมจากโค้ดได้ ซึ่งจะช่วยเสริมความมั่นคงปลอดภัยโดยรวมของระบบ โดยมีแนวปฏิบัติดังนี้

(1) จัดเก็บ Source Code ให้ปลอดภัยไว้ในระบบจัดการ Source Code ในเครื่องคอมพิวเตอร์ที่ไม่ใช่เครื่อง Production เช่น Git หรือใช้บริการ Azure DevOps Repos ของมหาวิทยาลัย หรือใช้บริการภายนอก เช่น GitHub หรือ GitLab เป็นต้น

(2) ใช้กระบวนการ Deployment ไปยัง Production Web Server ที่เหมาะสม แทนการแทนที่จะคัดลอก Source Code ทั้งหมดไปวางบนเครื่องคอมพิวเตอร์แม่ข่าย เช่น การใช้เครื่องมือ Web Publishing, การใช้เครื่องมือ CI/CD หรือการใช้ Docker Container เป็นต้น

(3) ลบไฟล์ที่ไม่จำเป็นออกจาก Production Web Server เช่น .git directory หรือไฟล์ที่ใช้ชั่วคราวระหว่างการ test และ build เป็นต้น

5. ด้าน Secure Sensitive Data

5.1 ใช้บริการ Login ของมหาวิทยาลัยที่สนับสนุน Multi-Factor Authentication ในการเข้าสู่ระบบเสมอ

การบังคับใช้การเข้าสู่ระบบผ่านบริการ Login ของมหาวิทยาลัยที่รองรับ Multi-Factor Authentication (MFA) จะช่วยเพิ่มความปลอดภัยอีกชั้น นอกเหนือจากรหัสผ่าน โดยกำหนดให้ยืนยันตัวตนด้วยวิธีที่สอง เช่น OTP หรือแอป authenticator ทำให้ผู้ไม่หวังดีเข้าถึงบัญชีได้ยากขึ้นมากแม้จะรู้รหัสผ่าน

5.2 ทำ Data Masking, Anonymous Data และ Redaction สำหรับการส่งออกข้อมูลส่วนบุคคลเสมอ

เมื่อต้องส่งออกหรือแบ่งปันข้อมูลส่วนบุคคล ควรทำ Data Masking (การปกปิด), Anonymization (การทำนิรนาม) หรือ Redaction (การตัดทอน) เพื่อแปลงหรือลบข้อมูลที่ระบุตัวตนได้ ทำให้ข้อมูลไม่สามารถย้อนกลับไปหาเจ้าของได้โดยตรง ซึ่งช่วยปกป้องความเป็นส่วนตัวและลดความเสี่ยงจากการรั่วไหลของข้อมูลส่วนบุคคลและข้อมูลที่เป็นความลับ โดยมีแนวปฏิบัติดังนี้

(1) แทนที่ข้อมูล (Data Masking) ที่เป็นความลับทั้งหมดหรือบางส่วนด้วยอักษรภาษาอังกฤษ เช่น แทนที่เลขบัตรประชาชน 5 หลักสุดท้ายด้วยตัวอักษร X ห้าตัว ตัวอย่าง “36599000XXXXX”

(2) ใช้ข้อมูลไม่ระบุตัวตน (Anonymous Data) ในการวิเคราะห์ข้อมูลหรือเผยแพร่สถิติ เพื่อป้องกันการเปิดเผยข้อมูลส่วนบุคคลโดยไม่จำเป็น

(3) ซ่อนข้อความ (Remove Text Redaction) บนเอกสารรูปแบบไฟล์ PDF ก่อนนำไปเผยแพร่ โดยใช้ซอฟต์แวร์แก้ไขไฟล์ PDF เช่น Adobe Acrobat หรือ Foxit PDF Editor

(4) กำหนดระยะเวลาในการเก็บ รวบรวม ใช้ หรือเปิดเผยข้อมูลที่เผยแพร่บนเว็บไซต์

5.3 ไม่เก็บรหัสผ่าน กุญแจลับ (secret key) หรือ API key ที่สามารถเข้าถึงได้ในโฟลเดอร์เก็บเว็บไซต์

การไม่เก็บรหัสผ่าน, Secret Key หรือ API Key ในโฟลเดอร์เก็บเว็บไซต์โดยตรงเป็นสิ่งจำเป็น เพื่อป้องกันการเข้าถึงข้อมูลที่ละเอียดอ่อนเหล่านี้เมื่อเครื่องคอมพิวเตอร์แม่ข่ายถูกบุกรุก ทำให้ผู้โจมตีไม่สามารถนำข้อมูลไปใช้ในการเข้าถึงระบบหรือบริการอื่น ๆ ได้ ช่วยลดความเสียหายอย่างร้ายแรงต่อระบบและความลับของข้อมูล โดยมีแนวปฏิบัติดังนี้

(1) แยกข้อมูลละเอียดอ่อนออกจาก Source Code และ Web Root เสมอ
(2) ใช้ Environment Variables (ตัวแปรสภาพแวดล้อม) ในการจัดการ Secrets ในสภาพแวดล้อม Production
(3) สำหรับระบบที่มีความซับซ้อนและต้องการความปลอดภัยสูงขึ้น ควรพิจารณาใช้บริการเฉพาะทาง เช่น Azure Key Vault

6. ด้าน Response and Recovery

6.1 เปิดการจัดเก็บข้อมูลจราจรของ Web Server (Log file) เป็นระยะเวลาอย่างน้อย 90 วัน

การเปิดใช้งานและจัดเก็บข้อมูลจราจร (Log File) ของ Web Server เป็นระยะเวลาอย่างน้อย 90 วัน มีความสำคัญอย่างยิ่งต่อการสืบสวนเหตุการณ์ด้านความมั่นคงปลอดภัย ช่วยให้สามารถตรวจจับการโจมตี ระบุพฤติกรรมที่ผิดปกติ และใช้เป็นหลักฐานในการวิเคราะห์หาสาเหตุและขอบเขตความเสียหายของเหตุการณ์ที่เกิดขึ้นได้อย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) ทำการจัดเก็บ Log อย่างปลอดภัยโดยเก็บ Log File ไว้ใน Directory ที่อยู่นอก Web Root ของแอปพลิเคชัน และมีสิทธิ์การเข้าถึง (File Permissions) ที่จำกัด เพื่อป้องกันการเข้าถึงจากภายนอก
(2) ตั้งค่าให้มีการหมุนเวียน Log อัตโนมัติ (เช่น ทุกวัน, ทุกสัปดาห์) โดยบีบอัดไฟล์เก่าและลบไฟล์ที่เกินระยะเวลาจัดเก็บ อย่างน้อย 90 วัน
(3) ตรวจสอบ Log ที่จัดเก็บเป็นระยะ เช่น ทุก 1-3 เดือน เพื่อให้มั่นใจว่าระบบได้ทำการจัดเก็บ Log อย่างต่อเนื่องและไฟล์มีความเป็นปัจจุบัน

6.2 มีการทดสอบช่องโหว่ความปลอดภัยของเว็บไซต์และเว็บแอปพลิเคชันอย่างน้อยทุก 3-6 เดือน

ควรมีการทดสอบช่องโหว่ความปลอดภัยของเว็บไซต์และเว็บแอปพลิเคชันอย่างสม่ำเสมออย่างน้อยทุก 3-6 เดือน เพื่อค้นหาและแก้ไขจุดอ่อนที่อาจถูกโจมตีได้ เช่น ช่องโหว่ SQL Injection หรือ XSS ซึ่งจะช่วยลดความเสี่ยงจากการถูกบุกรุกและรักษาความมั่นคงปลอดภัยของข้อมูลระบบได้อย่างต่อเนื่อง โดยมีแนวปฏิบัติดังนี้

(1) กำหนดความถี่ในการทดสอบที่เหมาะสม เช่น อย่างน้อยทุก 3-6 เดือน หรือความถี่มากขึ้นสำหรับระบบที่มีความเสี่ยงสูง หรือ ทุกครั้งที่มีการเปลี่ยนแปลงสำคัญ
(2) ใช้เครื่องมือและวิธีการที่หลากหลาย เช่น Vulnerability Scanners หรือ Manual Penetration Testing โดยใช้ซอฟต์แวร์ เช่น OWASP ZAP, Burp Suite, Nessus, Acunetix และ Qualys เป็นต้น
(3) มีการทำ source code review เพื่อหา code ที่ไม่ปลอดภัย หรือ การใช้ component หรือ library ที่มีช่องโหว่ด้านความปลอดภัย เพื่อทำการแก้ไขหรืออัปเดต ก่อนนำไปใช้งานจริง

6.3 แก้ไขปัญหาด้านความปลอดภัยที่ได้รับแจ้งจากศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์ให้ครบถ้วน

การตอบสนองและแก้ไขปัญหาด้านความมั่นคงปลอดภัยที่ได้รับแจ้งจากศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์เป็นสิ่งสำคัญยิ่ง ต้องดำเนินการให้ครบถ้วนและทันท่วงที เพื่อปิดช่องโหว่ที่ถูกเปิดเผย ลดความเสี่ยงที่ระบบจะถูกโจมตี และป้องกันความเสียหายที่อาจเกิดขึ้นได้อย่างมีประสิทธิภาพ โดยมีแนวปฏิบัติดังนี้

(1) แจ้งช่องทางการรับข้อมูลข่าวสาร เช่น e-mail และเบอร์โทรศัพท์ ให้กับศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์จัดเก็บข้อมูลไว้ เพื่อใช้ในการแจ้งปัญหาด้านความปลอดภัย และควรมีการแจ้งการเปลี่ยนแปลงข้อมูลทุกครั้งหากมีการเปลี่ยน e-mail หรือเบอร์โทรศัพท์ หรือมีการเปลี่ยนแปลงผู้รับผิดชอบ

(2) ดำเนินการแก้ไขปัญหาอย่างทันท่วงทีและครบถ้วน เมื่อได้รับการแจ้งจากศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์ รวมทั้งดำเนินการตามมาตรการป้องกันการเกิดปัญหาซ้ำ ตามคำแนะนำที่ได้รับ

(3) ประสานงานและรายงานกลับไปยังศูนย์รักษาความมั่นคงปลอดภัยไซเบอร์ เมื่อมีการดำเนินการแล้วเสร็จ หรือพบปัญหาอุปสรรคใด ๆ ที่ทำให้ไม่สามารถดำเนินการต่อได้

ภาคผนวก ก แหล่งข้อมูลเพิ่มเติม

ข้อ	แนวปฏิบัติ	แหล่งข้อมูล
1.1	ปิดบริการที่ไม่ปลอดภัย หรือไม่จำเป็นต่อการให้บริการเว็บไซต์และเว็บแอปพลิเคชัน	ระบบปฏิบัติการ Windows Server : https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/windows-firewall-with-advanced-security Linux Firewall https://help.ubuntu.com/community/UFW หรือ iptables https://www.netfilter.org/
1.4	อัปเดตแพตช์ซอฟต์แวร์ระบบปฏิบัติการและซอฟต์แวร์บริการอื่น ๆ ที่ใช้ให้เป็นปัจจุบันอยู่เสมอ	Linux Ubuntu https://help.ubuntu.com/community/AptGet/FullUpgrade Linux Fedora https://docs.fedoraproject.org/en-US/fedora/latest/system-administrators-guide/Package_Management_and_Software_Updates/Working_with_DNF/
1.5	ตั้งเวลาของเครื่องคอมพิวเตอร์แม่ข่ายโดยใช้มาตรฐานเวลาตามที่ส่วนกลางกำหนดไว้	Linux Ubuntu : https://www.digitalocean.com/community/tutorials/how-to-set-up-time-synchronization-on-ubuntu-20-04 Linux Fedora: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/latest/html/configuring_time_synchronization/introduction-to-chrony-suite
2.3	ปิดการใช้งาน HTTP (Hyper Text Transfer Protocol)	Apache mod_rewrite documentation: https://httpd.apache.org/docs/current/mod/mod_rewrite.html Redirect HTTP to HTTPS in IIS with URL Rewrite: https://learn.microsoft.com/en-us/iis/extensions/url-rewrite-module/creating-rewrite-rules-for-the-url-rewrite-module NGINX HTTP to HTTPS Redirect: https://nginx.org/en/docs/http/ngx_http_core_module.html#return Certbot User Guide (สำหรับวิธีการจัดการ Certbot): https://certbot.eff.org/docs/userguide.html
2.4	เปิดการใช้งาน HSTS (HTTP Strict Transport Security)	HSTS (Strict-Transport-Security) configuration for Apache: https://hstspreload.org/ (ดูส่วนการตั้งค่าสำหรับ Apache) Adding Custom HTTP Response Headers in IIS: https://learn.microsoft.com/en-us/iis/configuration/system.webserver/httpprotocol/customheaders NGINX add header documentation: https://nginx.org/en/docs/http/ngx_http_headers_module.html#add_header
2.5	ปิดการใช้งาน Algorithm การเข้ารหัส HTTPS ที่ไม่ปลอดภัย	ระบบปฏิบัติการ Windows: https://techpress.net/how-to-disable-tls-1-0-and-tls-1-1-on-windows-servers/

ข้อ	แนวปฏิบัติ	แหล่งข้อมูล
		ซอฟต์แวร์ Apache: https://httpd.apache.org/docs/current/mod/mod_ssl.html ซอฟต์แวร์ Nginx: http://nginx.org/en/docs/http/ngx_http_ssl_module.html
2.6	กำหนดสิทธิ์การเข้าถึงไฟล์ (file) และโฟลเดอร์ (folder) อย่างปลอดภัยตามสิทธิ์การใช้งาน	Apache HTTP Server Documentation - Module mod_autoindex: https://httpd.apache.org/docs/current/mod/mod_autoindex.html Configure Directory Browse in IIS: https://learn.microsoft.com/en-us/iis/configuration/system.webserver/directorybrowse How to Disable Directory Listing in NGINX: https://www.nginx.com/blog/how-to-disable-directory-listing-nginx
4.15	ควบคุมการทำดัชนีการค้นหาของบริการค้นหา (Search Engine)	Google Search Central: What is a robots.txt file?: https://developers.google.com/search/docs/crawling-indexing/robots/intro Google Search Central: robots.txt specifications: https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt Google Search Console: https://search.google.com/search-console (สำหรับ Robots.txt Tester)
6.1	เปิดการจัดเก็บข้อมูลจราจรของ Web Server (Log file) เป็นระยะเวลาอย่างน้อย 90 วัน	Mastering IIS Logging: Essential Basics: https://openobserve.ai/articles/iis-logging-basics Apache Logging Basics - The Ultimate Guide To Logging: https://www.loggly.com/ultimate-guide/apache-logging-basics How to Check & Configure NGINX Access & Error Logs: https://sematext.com/blog/nginx-logs
6.2	มีการทดสอบช่องโหว่ความปลอดภัยของเว็บไซต์และเว็บแอปพลิเคชันอย่างน้อยทุก 3-6 เดือน	ZAP (open source): https://www.zaproxy.org/getting-started Burp Suit (Free and paid version): https://portswigger.net/burp/documentation/desktop Metasploit (Free and paid version): https://docs.metasploit.com

ภาคผนวก ข กฎหมายและมาตรฐานสากลที่เกี่ยวข้อง

กฎหมาย มาตรฐานและแนวปฏิบัติที่เกี่ยวข้อง

- พระราชบัญญัติการรักษาความมั่นคงปลอดภัยไซเบอร์ พ.ศ. 2562 (Cybersecurity Act)
- พระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล พ.ศ. 2562 (PDPA)
- มาตรฐาน ISO/IEC 27001:2022 ฉบับภาษาไทย โดย ดร. บรรจง หะรังษี ผู้แปลและเรียบเรียง ระบบบริหารจัดการความมั่นคงปลอดภัยสารสนเทศ (ISMS)
- NIST SP 800-44 Rev.2 – Securing Public Web Servers
- แนวปฏิบัติการรักษาความมั่นคงปลอดภัยเว็บไซต์ (Website Security Guideline) โดย สำนักงานคณะกรรมการการรักษาความมั่นคงปลอดภัยไซเบอร์แห่งชาติ (สกมช.)
- OWASP Top 10 Security Risks
- Mozilla Web Security Guidelines