มาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสาร
สำหรับธุรกรรมภาษีทางอิเล็กทรอนิกส์

ว่าด้วยความมั่นคงปลอดภัยสารสนเทศ สำหรับโปรแกรมประยุกต์บนเว็บ

RD ICT Standard for Electronic Tax Transactions
: Information Security for Web Applications

RD STD. [06-2566]

กรมสรรพากร

Version 01.00.0000

คำนำ

กรมสรรพากร มีการพัฒนาและยกระดับหน่วยงานให้สอดคล้องกับทิศทางการขับเคลื่อนของประเทศ ตามแผนพัฒนารัฐบาลดิจิทัลของประเทศไทย ระบบ Web Application เป็นระบบที่จำเป็นต่อการพัฒนารัฐบาลดิจิทัล เพื่อให้การปฏิบัติงานและการให้บริการของหน่วยงานมีประสิทธิภาพ ปัจจุบัน กรมสรรพากร ได้พัฒนาและใช้งาน Web Application เพื่อให้บริการประชาชนในด้านการจัดเก็บภาษี รวมถึงระบบงานที่ใช้ภายในหน่วยงาน ดังนั้น กรมสรรพากร จึงได้จัดทำมาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสารว่าด้วยความมั่นคงปลอดภัยของ Web Application เพื่อให้การให้บริการของกรมสรรพากรบนระบบ Web Application เกิดความมั่นคงปลอดภัยตามมาตรฐานสากล

ประวัติการปรับปรุงเอกสาร

Version	รายละเอียด	วันที่
01.00.0000	เวอร์ชันแรก	23 สิงหาคม 2566

สารบัญ

เรื่อง **หน้าที่**

1. ขอบข่าย

มาตรฐานการดำเนินการเกี่ยวกับความมั่นคงปลอดภัย สำหรับโปรแกรมประยุกต์บนเว็บ ฉบับนี้ประกอบด้วย รายการมาตรฐานที่ใช้ในพัฒนา Web Application ให้มีความมั่นคงปลอดภัย เช่น การใช้งาน Library และ Framework ที่ได้รับการยอมรับว่ามีความปลอดภัยสูงตามการจัดอันดับช่องโหว่ที่พบบ่อย 10 อันดับ (OWASP Top 10) เป็นต้น

มาตรฐานความมั่นคงปลอดภัย สำหรับโปรแกรมประยุกต์บนเว็บ มีข้อกำหนด 4 ด้าน ดังนี้

(1) การวางแผนการพัฒนา Web Application

(2) การพัฒนา Web Application ให้มีความมั่นคงปลอดภัย

(3) การตรวจสอบช่องโหว่และปิดช่องโหว่ของระบบ Web Application

(4) การบำรุงรักษาระบบ Web Application

2. นิยาม

ความหมายของคำที่ใช้ในมาตรฐานด้านความมั่นคงปลอดภัย สำหรับโปรแกรมประยุกต์บนเว็บ มีดังต่อไปนี้

1.1 ผู้พัฒนาระบบ หมายถึง เจ้าหน้าที่กรมสรรพากร หรือเจ้าหน้าที่ผู้ที่กรมสรรพากรทำสัญญาว่าจ้าง (Outsource) ที่ได้รับมอบหมายให้พัฒนา Web Application

1.2 การเข้ารหัสข้อมูล หมายถึง กระบวนการแปลงข้อความธรรมดาเป็นข้อความที่เข้ารหัสเพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต

1.3 สิทธิของผู้ใช้งาน หมายถึง สิทธิทั่วไป สิทธิจำเพาะ สิทธิพิเศษ และสิทธิอื่นใดที่เกี่ยวข้องกับระบบ Web Application

1.4 ช่องโหว่ หมายถึง ช่องทาง หรือจุดอ่อนอย่างหนึ่งที่ ทำให้ผู้โจมตีสามารถเข้าถึงข้อมูลได้ง่าย

1.5 การพิสูจน์ตัวตน หมายถึง ขั้นตอนที่ตรวจสอบหลักฐานเพื่อแสดงว่าเป็นบุคคลที่กล่าวอ้างจริง

3. การวางแผนการพัฒนา Web Application

การวางแผนพัฒนา Web Application ต้องคำนึงถึงความปลอดภัยระหว่างการออกแบบและพัฒนา รวมถึงแผนการทดสอบและปรับปรุงเพื่อแก้ไขปัญหาที่พบ ตามกระบวนการ Secure Software Development Lifecycle (SSDLC) ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์ที่ให้ความสำคัญกับการรักษาความปลอดภัยของซอฟต์แวร์ตั้งแต่เริ่มต้นการพัฒนาจนถึงการเผยแพร่

การวางแผนพัฒนาแบบ DevSecOps เป็นการผสมผสานระหว่างระบวนการพัฒนาซอฟต์แวร์ (SSDLC) และทฤษฎี DevOps เพื่อเพิ่มประสิทธิภาพและความปลอดภัยของการพัฒนาระบบ มีการนำเทคโนโลยีต่าง ๆ เข้ามาช่วยในการตรวจสอบความปลอดภัยของระบบอย่างรวดเร็วและตลอดระยะเวลาของ SSDLC ซึ่งมีหลักเกณฑ์ดังต่อไปนี้

1) ขั้นตอนแนวคิด (Concept) การวิเคราะห์แนวคิดและข้อกำหนดของระบบ จะต้องพิจารณาเกี่ยวกับการรักษาความปลอดภัยของระบบและการเข้ารหัสข้อมูลที่ส่งผ่านระบบ

2) ขั้นตอนออกแบบ (Design) การพิจารณาเกี่ยวกับวิธีการใช้เทคโนโลยีต่าง ๆ เพื่อช่วยในการตรวจสอบความปลอดภัยของระบบอย่างรวดเร็วและมีประสิทธิภาพ

3) ขั้นตอนการพัฒนา (Development) การใช้เทคนิคต่าง ๆ เช่น Shift Left Testing และ Continuous Integration and Continuous Delivery (CI/CD) เพื่อช่วยในการตรวจสอบความปลอดภัยของระบบในแต่ละขั้นตอนของ SDLC

4) ขั้นตอนทดสอบ (Testing) การทดสอบระบบทั้งหมดเพื่อตรวจสอบความถูกต้องและความปลอดภัย รวมถึงการเรียกใช้หลักการการทดสอบแบบ Automated Testing ในการตรวจสอบความถูกต้องของระบบ

5) ขั้นตอนการประเมินและการตรวจสอบ (Assessment) การตรวจสอบความปลอดภัยของระบบโดยใช้เทคนิคต่าง ๆ เพื่อตรวจสอบความเสี่ยงและช่องโหว่ที่อาจเกิดขึ้น การทดสอบนี้จะมีเป้าหมายเพื่อตรวจหาช่องโหว่ (Vulnerabilities) และข้อบกพร่อง (Flaws) ที่อาจทำให้ผู้ไม่ประสงค์ดีเข้าถึงระบบได้

4. การพัฒนา Web Application ให้มีความมั่นคงปลอดภัย

4.1 การพัฒนาโปรแกรม Web Application ให้มีความมั่นคงปลอดภัย จะต้องใช้เทคนิคที่ถูกต้องและปลอดภัย เพื่อป้องกันการเข้าถึงข้อมูลโดยผู้ไม่มีสิทธิ ในการพัฒนา Web Application ให้มีความมั่นคงปลอดภัยและการปฏิบัติตามหลักการที่เป็นมาตรฐานเพื่อป้องกันช่องโหว่ต่าง ๆ จะต้องเป็นไปตามหลักเกณฑ์ดังต่อไปนี้

4.1.1 Input Validation หมายถึง การตรวจสอบความถูกต้องของข้อมูลที่รับเข้ามาในระบบ เพื่อป้องกันการโจมตีด้วยเทคนิค Injection หรือการแทรกโค้ดลงในระบบ

4.1.2 Output Encoding หมายถึง การแปลงข้อมูลเป็นรูปแบบที่ปลอดภัย เพื่อป้องกันการโจมตีด้วยเทคนิค Cross-site Scripting (XSS) หรือการแทรกโค้ดลงในหน้าเว็บไซต์

4.1.3 Authentication and Access Control หมายถึง การตรวจสอบและกำหนดสิทธิการเข้าถึงข้อมูลในระบบ เพื่อให้เฉพาะผู้ที่มีสิทธิเท่านั้นที่จะเข้าถึงข้อมูลในระบบ

4.1.4 Password Management หมายถึง การจัดการรหัสผ่านอย่างปลอดภัย เพื่อลดความเสี่ยงในการเข้าถึงระบบโดยผู้ไม่มีสิทธิ

4.1.5 Error Handling and Logging การจัดการและบันทึกข้อผิดพลาดในระบบ เพื่อให้ผู้ดูแลระบบสามารถตรวจสอบแก้ไขปัญหาได้อย่างรวดเร็วและถูกต้อง

4.1.6 Cryptography หมายถึง การใช้เทคโนโลยีการเข้ารหัสและถอดรหัสข้อมูล เพื่อป้องกันการเข้าถึงข้อมูลโดยไม่มีสิทธิ

4.2 การลดช่องโหว่ที่อาจเกิดขึ้นในกระบวนการการพัฒนา Web Application ตามการจัดอันดับช่องโหว่ที่พบบ่อย 10 อันดับ (OWASP Top 10) จะต้องเป็นไปตามหลักเกณฑ์ดังต่อไปนี้

4.2.1 Broken Access Control หมายถึง ปัญหาจากการจัดการสิทธิการเข้าถึงข้อมูลและระบบของผู้ใช้งานในแอปพลิเคชัน ซึ่งอาจจะเกิดจากการตั้งค่าสิทธิการเข้าถึงไม่ถูกต้อง หรือการที่ไม่มีการตรวจสอบสิทธิการเข้าถึงอย่างเหมาะสม จนทำให้ผู้ไม่มีสิทธิเข้าถึงระบบหรือข้อมูลสำคัญได้ มีแนวทางในการป้องกัน ดังนี้

1) กำหนดนโยบายการเข้าถึงข้อมูลที่เหมาะสม

2) ตรวจสอบสิทธิการเข้าถึงและการทำรายการในทุกส่วนของเว็บแอปพลิเคชัน

3) การตรวจสอบความถูกต้องของข้อมูลส่วนตัว และการใช้งาน Access Control List (ACL) ในการจัดการสิทธิการเข้าถึงข้อมูลและการทำรายการ

4.2.2 Cryptographic Failures คือ ช่องโหว่ที่เกี่ยวกับการใช้งานการเข้ารหัสของข้อมูลและการใช้งานการรักษาความลับในแอปพลิเคชันเว็บ ซึ่งส่งผลให้ผู้ไม่ประสงค์ดีสามารถเข้าถึงข้อมูลได้โดยไม่ได้รับอนุญาต มีแนวทางในการป้องกัน ดังนี้

1) ใช้งานอัลกอริทึมสำหรับเข้ารหัสที่เหมาะสม การเพิ่มความยากในการเดาและการถอดรหัส

2) ใช้งานรหัสผ่านที่มีความซับซ้อน และเปลี่ยนรหัสผ่านอย่างสม่ำเสมอ

3) ใช้งานโมดูลการเข้ารหัสที่ได้รับการตรวจสอบและรับรองว่าปลอดภัย

4) ใช้งาน HTTPS ในการส่งข้อมูลระหว่างเว็บแอปพลิเคชันกับผู้ใช้

5) ใช้งานการเข้ารหัสที่มีความปลอดภัยสูงและได้รับการยอมรับ

4.2.3 Injection คือ ช่องโหว่ที่เกี่ยวกับการป้อนข้อมูลที่ผิดปกติเข้าไปในแอปพลิเคชัน โดยที่ข้อมูลดังกล่าวมักอยู่ในรูปแบบคำสั่ง ที่มีความมุ่งเน้นไปที่การเข้าถึงฐานข้อมูล หรือการแก้ไขการทำงานของระบบ การป้องกันการโจมตีด้วย Injection มีแนวทางการป้องกัน ดังนี้

1) การใช้ Parameterized Statements หรือ Prepared Statements ช่วยป้องกันไม่ให้ผู้ไม่ประสงค์ดีสามารถ Inject SQL Query เข้ามาในระบบ

2) การใช้ Stored Procedures คือ การเขียนโค้ดภายในฐานข้อมูล ซึ่งมีความปลอดภัยมากกว่าการรันคำสั่ง SQL Query ผ่าน Application

3) การทำ Input Validation เป็นการตรวจสอบค่าข้อมูล Input ว่ามีรูปแบบถูกต้องตามที่กำหนดไว้หรือไม่

4) การใช้ Framework ที่มีการป้องกัน Injection ใช้ Framework ที่รวมการป้องกัน Injection เข้าไว้ในตัวเอง

5) ไม่เขียนคำสั่ง SQL โดยตรงในตัวแปร (Parameter) ที่ส่งโดยตรงไปยังโปรแกรมประยุกต์บนเว็บ

6) ควบคุมการแสดงผลข้อมูล Error Message

7) กำหนดสิทธิขั้นต่ำให้บัญชีผู้ใช้ของฐานข้อมูล

4.2.4 Insecure Design คือ การออกแบบระบบที่ไม่มีความปลอดภัย หรือมีข้อบกพร่องในการออกแบบที่ทำให้ระบบมีความเสี่ยงต่อการโจมตีได้ง่าย มีแนวทางในการป้องกัน ดังนี้

1) การออกแบบระบบโดยคำนึงถึงความปลอดภัยในทุกระดับ

2) ต้องมีการกำหนดสิทธิของผู้ใช้งาน

3) ต้องมีการตรวจสอบและควบคุมความเชื่อถือได้ของข้อมูล

4) มีการควบคุมการเข้าถึงและการใช้งานของข้อมูลโดยเหมาะสม

5) ต้องมีการอัพเดทและเพิ่มประสิทธิภาพในการป้องกันช่องโหว่และความไม่ปลอดภัยอยู่เสมอ

4.2.5 Security Misconfiguration คือ ช่องโหว่ที่เกิดจากการตั้งค่าไม่ถูกต้องหรือการปรับแต่งค่าที่ไม่เหมาะสมของระบบทำให้เกิดช่องโหว่ด้านความมั่นคงปลอดภัยของระบบ มีแนวทางในการป้องกัน ดังนี้

1) การตรวจสอบการกำหนดค่าอย่างเหมาะสม ตรวจสอบการกำหนดค่าในระบบและแอปพลิเคชันทั้งหมด เพื่อให้แน่ใจว่าค่าต่าง ๆ ถูกกำหนดอย่างเหมาะสม

2) การปิดใช้งานฟังก์ชันที่ไม่จำเป็น ทำการปิดใช้งานฟังก์ชันที่ไม่จำเป็นและสามารถเป็นช่องโหว่ได้

3) การอัปเดตซอฟต์แวร์ อัปเดตซอฟต์แวร์เพื่อป้องกันช่องโหว่ที่อาจเกิดขึ้นจากการใช้งานซอฟต์แวร์เก่า

4) การตรวจสอบการตั้งค่าความปลอดภัย ตรวจสอบการตั้งค่าความปลอดภัยบนระบบและแอปพลิเคชันทั้งหมด เช่น การตั้งค่าการเข้ารหัส การตั้งค่าและการกำหนดสิทธิการเข้าถึงให้กับผู้ใช้งาน

4.2.6 Vulnerable and Outdated Components คือ ช่องโหว่ที่เกิดจากการใช้งานส่วนประกอบของซอฟต์แวร์ที่มีช่องโหว่ หรือใช้เวอร์ชันเก่าที่มีช่องโหว่ มีแนวทางในการป้องกัน ดังนี้

1) การตรวจสอบและอัปเดตระบบ ตรวจสอบระบบอย่างสม่ำเสมอเพื่อตรวจหาช่องโหว่ที่เกิดจากโปรแกรมที่ติดตั้ง และอัปเดตให้เป็นเวอร์ชันล่าสุด โดยอัปเดตทั้งโปรแกรมและไลบรารีที่ใช้งาน

2) การตรวจสอบไฟล์ ตรวจสอบไฟล์ทั้งหมดที่ใช้งานในระบบเพื่อค้นหาไฟล์ที่มีช่องโหว่

3) การใช้งานช่องทางการอัปเดต การใช้ช่องทางการอัปเดตอย่างเหมาะสมให้กับระบบและโปรแกรมที่ใช้งาน เพื่อให้การอัปเดตเวอร์ชันมีความปลอดภัย

4) การตรวจสอบและปิดใช้งานช่องโหว่ที่พบ การตรวจสอบช่องโหว่และปิดใช้งานไฟล์ที่ไม่จำเป็น บริการและโปรแกรมที่ไม่ได้ใช้งาน

4.2.7 Identification and Authentication Failures คือ ปัญหาในการตรวจสอบและรับรองตัวตนของผู้ใช้ ซึ่งส่งผลให้ผู้ไม่หวังดีสามารถเข้าถึงและใช้งานแอปพลิเคชัน มีแนวทางในการป้องกัน ดังนี้

1) ใช้การรับรองความถูกต้องหลายปัจจัย (multi-factor authentication) เพื่อเพิ่มความปลอดภัยในการตรวจสอบตัวตนของผู้ใช้งาน

2) การประยุกต์ใช้การรับรองตัวตนระดับสูง (strong authentication) เช่น การใช้เทคโนโลยีการรับรองตัวตนด้วยใบหน้า (facial recognition) หรือการสแกนลายนิ้วมือ (fingerprint scanning) เป็นต้น

3) การใช้นโยบายการกำหนดระดับสิทธิการเข้าถึง (access control policy) เพื่อจำกัดสิทธิการเข้าถึงของผู้ใช้งานในแต่ละระดับ

4) การอัปเดตและปรับปรุงระบบการรับรองตัวตนเป็นประจำ

5) การส่งเสริมการฝึกอบรมและการสอนการใช้งานระบบให้แก่พนักงาน เพื่อเพิ่มความเข้าใจในการใช้งานระบบให้ถูกต้องและปลอดภัย

6) การตรวจสอบการเข้าถึงระบบโดยตลอดเวลาและการทำ Logging เพื่อตรวจสอบการเข้าถึงและกิจกรรมของผู้ใช้งานในระบบ

7) การใช้เทคโนโลยีการรับรองตัวตนที่มีความปลอดภัยมากขึ้นเช่น SSO (Single Sign-On)

4.2.8 Software and Data Integrity Failures คือ การใช้ซอฟต์แวร์หรือส่วนเสริมที่ไม่ได้ตรวจสอบความน่าเชื่อถือก่อนนำมาพัฒนา จนทำให้แฮกเกอร์สามารถดักแก้ไขค่าใน HTTP Request แล้วไปประมวลผลในส่วนที่ไม่ควรจะได้รับอนุญาตได้ มีแนวทางในการป้องกัน ดังนี้

1) ไม่ใช้ซอฟต์แวร์ที่เป็นเวอร์ชันไม่เสถียร

2) เลือกใช้ Linux OS ที่มีความเสถียร เช่น Debian / Centos

3) ตรวจสอบความถูกต้องของ Hashes File ที่ Download มาทุกครั้งก่อนติดตั้ง

4) ใช้งาน Software Repository ที่เชื่อถือได้เท่านั้น

5) ห้ามไม่ให้มีการสั่ง Deserialization กับข้อมูลจากผู้ใช้หากเป็นไปได้

6) มีการทบทวนกระบวนการ CI/CD และการเปลี่ยน configuration โดยให้สิทธิเฉพาะบุคคลที่ควรเข้าถึงได้เท่านั้น ไม่ให้มีการแทรกคำสั่งที่อันตรายเข้ามาได้

4.2.9 Security Logging and Monitoring Failures คือ ช่องโหว่ที่เกี่ยวข้องกับการทำ Log และ Monitoring การทำงานของระบบ หากไม่มีการทำ Security Logging and Monitoring อย่างเหมาะสมอาจทำให้ไม่สามารถตรวจสอบและติดตามการเกิดเหตุการณ์ที่เป็นอันตราย มีแนวทางในการป้องกัน ดังนี้

1) ออกแบบโครงสร้างของระบบ ให้มีการทำ Security Logging and Monitoring ที่เหมาะสม

2) มีการทำ Logging และ Monitoring การทำงานของระบบอย่างเหมาะสม

3) ตรวจสอบและติดตามกิจกรรมของระบบ เพื่อป้องกันการโจมตีและความไม่มั่นคงปลอดภัยของระบบ

4) ควรมีการกำหนดมาตรการความปลอดภัยที่ชัดเจนสำหรับการทำ Logging และ Monitoring การทำงานของระบบ

4.2.10 Server-Side Request Forgery (SSRF) คือ ช่องโหว่ที่เกิดจากการโจมตีที่เป้าหมายอยู่บนฝั่งเซิร์ฟเวอร์ โดยผู้โจมตีจะใช้โปรแกรมควบคุมการเข้าถึงเว็บเซิร์ฟเวอร์ที่มีช่องโหว่เพื่อดึงข้อมูลที่อยู่ภายในเครื่อง มีแนวทางในการป้องกัน ดังนี้

1) อัปเดตโมดูลและไลบรารีของแอปพลิเคชันให้เป็นเวอร์ชันล่าสุด

2) ตรวจสอบค่าพารามิเตอร์ของการร้องขอ ว่ามีการร้องขอถึงแหล่งข้อมูลอื่น ๆ นอกเหนือจากแหล่งข้อมูลที่ควรจะเปิดเผยหรือไม่

3) สร้าง URL Whitelist เพื่อจำกัดการเรียกใช้งานเฉพาะ URL ที่มีความปลอดภัยเท่านั้น

4) ใช้ Reverse Proxy เพื่อกรองการเรียกใช้งาน URL และตรวจสอบว่ามีการเรียกใช้งาน URL ตามที่กำหนดไว้เท่านั้น

5. การตรวจสอบช่องโหว่และปิดช่องโหว่ของระบบ Web Application

การตรวจสอบช่องโหว่และปิดช่องโหว่ของระบบ เพื่อป้องกันการเกิดปัญหาเป็นส่วนสำคัญของการดูแลระบบสารสนเทศและความมั่นคงปลอดภัยขององค์กร จะต้องเป็นไปตามหลักเกณฑ์ดังต่อไปนี้

1) Penetration Testing เป็นการทดสอบเชิงลึกเพื่อตรวจหาช่องโหว่ที่อาจถูกโจมตีจากผู้ไม่ประสงค์ดี โดยทดสอบด้วยวิธีการโจมตีจริง เช่น SQL Injection, Cross-Site Scripting (XSS), File Inclusion หรือการสร้าง Malware เป็นต้น

2) Vulnerability Scanning เป็นการสแกนระบบเพื่อตรวจสอบช่องโหว่และข้อผิดพลาดของโปรแกรมและเว็บไซต์ โดยจะทำการสแกนเนื้อหาต่าง ๆ ในเว็บไซต์ เช่น ตรวจสอบรหัส HTTP ตรวจสอบ SSL Certificate และประเภทของซอฟต์แวร์ที่ใช้ เพื่อหาช่องโหว่ที่อาจเกิดขึ้น เช่น Cross-site scripting (XSS), SQL injection หรือ Remote Code Execution (RCE) เป็นต้น

3) Code Review คือ กระบวนการตรวจสอบโค้ดที่เขียนขึ้นมาโดยผู้อื่น เพื่อหาข้อผิดพลาดหรือช่องโหว่ในการเขียนโค้ด การทำ Code Review นั้นสามารถทำได้หลายวิธี เช่น การส่งโค้ดให้ผู้อื่นตรวจสอบ หรือการใช้เครื่องมือที่ช่วยตรวจสอบโค้ดอัตโนมัติ

4) Social engineering testing เป็นการทดสอบโดยการใช้เทคนิคการจับข้อเท็จจริงเพื่อขโมยข้อมูล หรือเข้าถึงระบบของ web application โดยการทำ social engineering testing นั้นจะใช้เทคนิคการชักชวน หรือการหลอกถามข้อมูล

6. การบำรุงรักษาระบบ Web Application

การดูแลและบำรุงรักษาระบบให้มีประสิทธิภาพและคงทนต่อการใช้งาน การบำรุงรักษาระบบจะต้องมีการประเมินและวิเคราะห์ข้อมูลเพื่อให้ทราบถึงปัญหาและข้อบกพร่องในระบบ และมีการสร้างแผนงานและกำหนดกิจกรรมที่จะช่วยในการแก้ไขปัญหา เพื่อเพิ่มประสิทธิภาพในด้านต่าง ๆ ของระบบ โดยจะต้องมีการจัดการทรัพยากรและการดูแลรักษาฮาร์ดแวร์ ซอฟต์แวร์ ฐานข้อมูล และรวมถึงดูแลการดำเนินการในด้านความปลอดภัยของระบบ นอกจากนี้ ยังต้องเน้นการเรียนรู้และการพัฒนาทักษะของผู้ดูแลระบบ เพื่อให้สามารถทำงานได้อย่างมีประสิทธิภาพ จะต้องเป็นไปตามหลักเกณฑ์ดังต่อไปนี้

1) การตรวจสอบช่องโหว่ (Vulnerability scanning) อย่างสม่ำเสมอช่วยลดความเสี่ยงที่จะเกิดการโจมตีในอนาคต ทำให้สามารถแก้ไขปัญหาได้ทันทีก่อนที่จะเกิดความเสียหาย อีกทั้งเป็นการตรวจสอบการเข้าถึงของผู้ไม่หวังดี

2) การทำสำรองข้อมูลที่สำคัญเพื่อให้สามารถกู้คืนข้อมูลได้กรณีเกิดเหตุการณ์ที่ไม่คาดคิด

3) การทบทวนสิทธิการเข้าใช้งานระบบเป็นประจำอย่างน้อยปีละ 1 ครั้ง โดยให้มีการกำหนดสิทธิเฉพาะบุคคล และไม่ให้มีผู้ไม่มีสิทธิเข้าถึงได้

4) การตรวจสอบว่าระบบได้ทำการเข้ารหัสข้อมูลที่สำคัญตามมาตรฐานความปลอดภัยหรือไม่ และมีการใช้เทคโนโลยีการเข้ารหัสที่เหมาะสมหรือไม่

5) การตรวจสอบและอัปเดตโค้ด (Code review and update) การตรวจสอบโค้ดเว็บแอปพลิเคชันและอัปเดตโค้ดเพื่อแก้ไขช่องโหว่ที่อาจเกิดขึ้นหลังจากการเผยแพร่

6) การตรวจสอบและอัปเดตแพตช์ระบบปฏิบัติการและโปรแกรมประยุกต์ต่าง ๆ (System patching) เพื่อป้องกันช่องโหว่ในระบบและโปรแกรมที่อาจเป็นช่องทางการเข้าถึงระบบให้กับผู้ไม่หวังดี

8) การตรวจสอบและบันทึกการเข้าถึง (Access monitoring and logging) ตรวจสอบการเข้าถึงข้อมูลและฟังก์ชันต่าง ๆ ซึ่งช่วยตรวจสอบการเข้าถึงของผู้ใช้งานว่ามีการกระทำไม่ถูกต้องหรือไม่

9) การสร้างความคิดริเริ่มทางด้านความปลอดภัย (Security awareness training) ฝึกอบรมผู้ใช้งานและพนักงาน เกี่ยวกับเทคนิคและมาตรการทางด้านความปลอดภัย เพื่อเพิ่มความคิดริเริ่มในการใช้งานระบบอย่างปลอดภัย

10) การสำรวจความเสี่ยง (Risk assessment) การตรวจสอบความเสี่ยงที่อาจเกิดขึ้นจากการโจมตีต่าง ๆ และใช้มาตรการลดความเสี่ยงดังกล่าว

บรรณานุกรม

[1] แนวนโยบายในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศของกรมสรรพากร พ.ศ. 2565

[2] แนวปฏิบัติในการรักษาความมั่นคงปลอดภัยด้านสารสนเทศของกรมสรรพากร พ.ศ. 2565

[3] ข้อเสนอแนะมาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสารที่จำเป็นต่อธุรกรรมทางอิเล็กทรอนิกส์ (ETDA Recommendation on ICT Standard for Electronic Transactions) ว่าด้วยมาตรฐานการรักษาความมั่นคงปลอดภัยสำหรับโปรแกรมประยุกต์บนเว็บ Web Application Security Standard ขมธอ. 4-2559 เวอร์ชัน 1.0

[4] ข้อกำหนดด้านความมั่นคงปลอดภัยสารสนเทศตามมาตรฐาน ISO/IEC 27002:2022 (Annex A : Information Security Controls Reference)

[5] คู่มือการตรวจสอบความมั่นคงปลอดภัยของเว็บแอปพลิเคชัน หรือ OWASP Testing Guide (OTG) V4

[6] เอกสารกรอบการทำงานเพื่อการรับประกันความมั่นคงปลอดภัยของซอฟต์แวร์ (OWASP SAMM)

[7] รายการปัญหาความมั่นคงปลอดภัยที่พบบ่อยที่สุดในการพัฒนาเว็บแอปพลิเคชัน ประจำปี ค.ศ. 2021 10 อันดับ (OWASP Top Ten 2021)