มาตรฐานรายการตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันเพื่อให้เป็นไปตามมาตรฐานการรักษาความมั่นคงปลอดภัยทางไซเบอร์

แบบฟอร์มตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
1	การพิสูจน์ตัวตนและการควบคุมการเข้าถึง (Authentication & Access Control) มาตรฐานที่อ้างอิง : OWASP 2021 (A01 และ A07) และ Zero Trust			
1.1	ใช้มาตรฐาน OAuth 2.0 หรือ OpenID Connect สำหรับการพิสูจน์ตัวตน เช่น การเข้าสู่ระบบด้วยบัญชี Google, Facebook หรือ ThaID เพื่อลดความเสี่ยงจากการจัดการรหัสผ่านโดยตรงและอำนวยความสะดวกให้ผู้ใช้			
1.2	พัฒนาระบบการยืนยันตัวตนหลายขั้นตอน (Multi-Factor Authentication: MFA) เช่น การใช้รหัสผ่านร่วมกับการยืนยันผ่านโทรศัพท์มือถือ (SMS OTP) ซึ่งจะช่วยลดความเสี่ยงได้ถึง 99.9% แม้รหัสผ่านจะถูกขโมย (อ้างอิงจาก NIST)			
1.3	กำหนดนโยบายรหัสผ่านที่รัดกุม โดยมีความยาวขั้นต่ำ 12 ตัวอักษร และประกอบด้วยตัวอักษรใหญ่-เล็ก ตัวเลข และสัญลักษณ์พิเศษเพื่อเพิ่มความปลอดภัยจากการโจมตีด้วยวิธี Brute Force Attack			
1.4	พัฒนาระบบการจัดการเซสชัน (Session Management) ของผู้ใช้ให้ปลอดภัย เช่น การเข้ารหัส Session ID และการกำหนดเวลาหมดอายุของเซสชันเพื่อป้องกันการเข้าถึงระบบโดยไม่ได้รับอนุญาต			

แบบฟอร์มตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	1.4.1 มีการเข้ารหัส Session ID			
	1.4.2 มีการกำหนด Session Timeout เช่น ตั้งค่าหมดอายุภายใน 30 นาทีเมื่อไม่ใช้งาน			
	1.4.3 มีการสร้าง Session ID แบบสุ่มที่คาดเดาไม่ได้			
	1.4.4 มีการส่ง Session ID ผ่านช่องทางที่เข้ารหัสลับ			
2	การเข้ารหัสข้อมูล (Data Encryption) มาตรฐานที่อ้างอิง : Cybersecurity Framework (NIST) และ Zero Trust			
2.1	เข้ารหัสข้อมูลที่สำคัญหรือข้อมูลที่จัดเก็บเพื่อให้สอดคล้องกับพระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคล (PDPA) (https://www.ratchakitcha.soc.go.th/DATA/PDF/2562/A/069/T_0052.PDF) มาตรา 37 และ 40 โดยใช้มาตรฐานเทียบเท่าหรือดีกว่า เช่น ข้อมูลทั่วไป : ใช้มาตรฐานขั้นต่ำ AES-128 หรือ RSA-2048 ข้อมูลอ่อนไหว (Sensitive Data) : ใช้มาตรฐาน AES-256 หรือ ECC-256			
2.2	ใช้โปรโตคอล TLS 1.2 ขึ้นไปเพื่อเข้ารหัสข้อมูลทุกครั้งที่มีการสื่อสารผ่านเครือข่าย เพื่อป้องกันการดักจับหรือแก้ไขข้อมูลโดยผู้ไม่หวังดี (Man-in-the-Middle Attacks) และสอดคล้องกับข้อกำหนดทางกฎหมายตามพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. 2560 มาตรา 5 และปิดการใช้งานโปรโตคอล			

แบบฟอร์มตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	TLS ที่ต่ำกว่า 1.2 ลงไปทั้งหมด			
3	การป้องกันภัยคุกคาม (Threat Protection) มาตรฐานที่อ้างอิง : OWASP 2021 (A03 และ A05) และ Cybersecurity Framework			
3.1	ป้องกันการโจมตีที่ใช้ช่องโหว่ของฐานข้อมูล (SQL Injection) เช่น ใช้ Prepared Statements, Stored Procedures หรือการเรียกใช้ข้อมูลผ่าน API			
3.2	พัฒนาระบบ Input Validation เช่น ตรวจสอบชนิด/รูปแบบ/ช่วงของข้อมูล เพื่อการตรวจสอบข้อมูลที่ผู้ใช้ป้อนเข้าสู่ระบบว่าถูกต้องตามรูปแบบที่คาดหวังหรือไม่ ก่อนที่จะนำข้อมูลนั้นไปประมวลผลต่อ ช่วยลดความเสี่ยงจากการโจมตีหลายรูปแบบ			
3.3	พัฒนาระบบ Output Validation เช่น การ Encoding Output, จำกัดข้อมูลที่ส่งออกเท่าที่จำเป็น			
3.4	พัฒนาระบบ Encoding เช่น เข้ารหัสข้อมูลที่รับส่งบนเครือข่ายโดยใช้ TLS 1.2 ขึ้นไปสำหรับรับส่งข้อมูลทั่วไป ใช้ AES-256-GCM กับข้อมูลที่เป็นความลับสูง			
3.5	ป้องกันการโจมตีที่แทรกโค้ดที่เป็นอันตรายลงในเว็บไซต์ (Cross-Site Scripting : XSS) เพื่อป้องกันการขโมย Session ของผู้ใช้			
3.6	ใช้งาน HTTPOnly Cookie flag บน Framework/Server			
3.7	พัฒนาระบบ Sanitization เช่น ใช้ Whitelisting แทน Blacklisting, ตรวจสอบ			

แบบฟอร์มตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
	อักขระพิเศษ (<>'"&), กำหนดความยาวสูงสุด (Max Length), ตรวจสอบว่าเป็นตัวเลขจริง, กำหนดช่วงค่า (Min-Max), ใช้ HTML Encoding ตรวจสอบว่าไม่มี JavaScript Code ปน, ตรวจสอบ MIME Type จริง (ไม่ใช่นามสกุลไฟล์), เปลี่ยนชื่อไฟล์ใหม่ (secure_filename), จำกัดขนาดไฟล์, หลีกเลี่ยง String Concatenation ใน SQL, แยก Command และ Arguments, ตรวจสอบ Input ก่อนส่งไปยัง Shell			
3.8	ป้องกันการโจมตีที่ใช้การปลอมแปลงคำขอจากผู้ใช้ (Cross-site request forgery : CSRF) โดยใช้ Token สำหรับยืนยันคำขอทุกครั้ง			
3.9	ใช้ Unique Token เช่น Cryptographic Random Token กำหนดอายุของ Token ระยะสั้น หรือใช้เครื่องมือ เช่น JWT (JSON Web Tokens), Redis			
3.10	ติดตั้งและใช้งานระบบ Captcha			
3.11	ตรวจสอบ Referrer เช่น กำหนดโดเมนที่อนุญาต, ตรวจสอบ Request ทุกรายการที่ไม่ใช่ GET หรือใช้ HTTS เท่านั้น			
3.12	ควบคุมการแสดงข้อความแจ้งเตือนและข้อผิดพลาด เช่น ไม่แสดงรายละเอียดระบบในข้อผิดพลาด เช่น Database Schema, Server Paths, ใช้ข้อความทั่วไปสำหรับผู้ใช้ แต่บันทึกรายละเอียดใน Log, ใช้ HTTP Status Code ที่ถูกต้อง เช่น 404, 500 เป็นต้น			

แบบฟอร์มตรวจสอบสำหรับผู้พัฒนาระบบสารสนเทศและแอปพลิเคชันตามมาตรฐานเพื่อการพัฒนาระบบสารสนเทศอย่างปลอดภัย				
ลำดับ	**หัวข้อ**	**ยอมรับได้**	**ยังต้องปรับปรุง**	**หมายเหตุ**
3.13	ไม่ใช้ Autocomplete ในฟอร์มที่สำคัญ			
3.14	หลีกเลี่ยงการใช้ URL ที่คาดเดาได้ง่าย			
4	การควบคุมการพัฒนาและทดสอบ (Development & Testing Controls) มาตรฐานที่อ้างอิง : OWASP 2021 (A04 และ A06)			
4.1	ใช้ Secure Software Development Life Cycle (SSDLC) ซึ่งเป็นกระบวนการพัฒนาซอฟต์แวร์ที่มีการพิจารณาความปลอดภัยในทุกขั้นตอน			
4.2	ทำการตรวจสอบโค้ดเพื่อหาช่องโหว่ด้านความปลอดภัย (Secure Code Review)			
4.3	ใช้เครื่องมือ Static Application Security Testing (SAST) เพื่อตรวจหาช่องโหว่			
5	การจัดการ Dependencies และ Components มาตรฐานที่อ้างอิง : OWASP 2021 (A06 และ A08)			
5.1	ตรวจสอบ Dependencies เพื่อหาช่องโหว่ โดยใช้เครื่องมือที่ใช้ในการตรวจสอบและจัดการส่วนประกอบของซอฟต์แวร์ หรือ Software Composition Analysis (SCA) Tools เช่น Dependency-Check			
5.2	จัดทำบัญชีรายการ Dependencies			
5.3	อัปเดต Libraries และ Components ทุก 3 เดือนหรือเมื่อพบช่องโหว่ร้ายแรง (CVSS >=7 (High และ Critical) อ้างอิงจาก CVSS v3.1)			