กรมควบคุมโรค
DEPARTMENT OF DISEASE CONTROL

(ร่าง)

นโยบายและแนวปฏิบัติ ด้านความมั่นคงปลอดภัยสารสนเทศ ด้านการพัฒนา จัดหาและดูแลระบบหรือแอปพลิเคชัน

(Secure Development Policy)

ประวัติการแก้ไขเอกสาร

ครั้งที่	วันที่	รายละเอียดการดำเนินการ
1	20 ม.ค. 2568	Draft Version 1
2	3 มี.ค. 2568	Draft Version 2
3	23 พ.ค. 2568	Draft Version 3

สารบัญ

1. วัตถุประสงค์

เพื่อให้การพัฒนาแอปพลิเคชันหรือซอฟต์แวร์ ให้มีความมั่นคงปลอดภัย ตลอดวงจรชีวิตของการพัฒนาเป็นไปอย่างปลอดภัย โดยยึดตามหลักการที่ได้ระบุไว้ในกรอบการพัฒนาตาม NIST Secure Software Development Framework (SSDF) และความมั่นคงปลอดภัยสารสนเทศตามมาตรฐาน ISO/IEC 27001

2. ขอบเขต

นโยบายนี้ใช้กับการพัฒนาแอปพลิเคชันหรือซอฟต์แวร์ทั้งหมดภายในกรมควบคุมโรค รวมทั้งการพัฒนาซอฟต์แวร์ของบุคคลที่สาม (Third Parties) และผู้ที่จ้างเหมาช่วง

3. บทบาทและความรับผิดชอบ

3.1. ฝ่ายบริหารระดับสูง

- ให้การสนับสนุนการริเริ่มพัฒนาอย่างมั่นคงปลอดภัย
- ให้การสนับสนุน สรรหาทรัพยากรให้เพียงพอสำหรับการดำเนินการด้านการพัฒนาระบบสารสนเทศที่มีความมั่นคงปลอดภัย

3.2. ทีมพัฒนาระบบ

- การปฏิบัติตามแนวปฏิบัติที่ได้กำหนดไว้ในนโยบายนี้

3.3. ทีมรักษาความมั่นคงปลอดภัยสารสนเทศ

- ให้คำแนะนำ และกำกับดูแลด้านการรักษาความมั่นคงปลอดภัยสารสนเทศ
- ดำเนินการประเมินความมั่นคงปลอดภัยสารสนเทศเป็นระยะ ๆ

3.4. ทีมรับรองคุณภาพ (QA / Tester)

- ดำเนินการตรวจสอบข้อกำหนดด้านความมั่นคงปลอดภัยสารสนเทศระหว่างดำเนินการทดสอบ
- กำกับดูแลให้มีการดำเนินการควบคุมด้านความปลอดภัยสารสนเทศให้เป็นไปตามแนวปฏิบัติที่ได้กำหนดไว้

4. นโยบายความมั่นคงปลอดภัยสารสนเทศด้านการพัฒนาแอปพลิเคชัน

4.1 นโยบาย และแนวปฏิบัติด้านการพัฒนาแอปพลิเคชันอย่างปลอดภัยตาม NIST

4.1.1. การเตรียมพร้อมด้านความมั่นคงปลอดภัยในการพัฒนา (PO)

4.1.1.1 ข้อกำหนดด้านความมั่นคงปลอดภัยในการพัฒนาซอฟต์แวร์ (PO.1)

นโยบาย

- เพื่อให้มีการระบุข้อกำหนดด้านความปลอดภัยสำหรับโครงสร้างพื้นฐาน การพัฒนาซอฟต์แวร์ขององค์กร และทบทวนข้อกำหนดให้เป็นปัจจุบัน
- เพื่อให้มีการจัดทำบันทึกข้อกำหนดด้านความปลอดภัยสำหรับการพัฒนาซอฟต์แวร์และทบทวนข้อกำหนดต่าง ๆ ให้เป็นปัจจุบัน
- เพื่อให้มีการสื่อสารข้อกำหนดด้านความปลอดภัยสำหรับการพัฒนาซอฟต์แวร์ให้บุคลากรและบุคคลภายนอกทราบ เพื่อนำกระบวนการพัฒนาซอฟต์แวร์ไปปฏิบัติ

แนวปฏิบัติ

- จัดทำข้อกำหนด ทบทวนและปรับปรุงข้อกำหนดด้านความปลอดภัยอย่างน้อยปีละ 1 ครั้ง หรือหากมีข้อกำหนดด้านความปลอดภัยใหม่ หรือเหตุการณ์ด้านความปลอดภัยที่สร้างความเสียหายที่อาจส่งผลกระทบกับระบบสารสนเทศ
- ให้ความรู้แก่บุคคลที่ได้รับผลกระทบเกี่ยวกับการเปลี่ยนแปลงข้อกำหนดที่มีการทบทวนและปรับปรุง
- การพัฒนาระบบให้คำนึงถึงความเสี่ยงที่เกิดขึ้น เช่น การสร้างซอสโค้ดแบบโมดูลาร์เพื่ออำนวยความสะดวกในการนำซอสโค้ดกลับมาใช้ซ้ำ และการอัปเดตโดยแยกส่วนประกอบด้านความปลอดภัยออกจากส่วนประกอบอื่นในระหว่างการดำเนินการเพื่อหลีกเลี่ยงคำสั่ง และการตั้งค่าที่ไม่ได้บันทึกไว้
- วิเคราะห์ความเสี่ยงของเทคโนโลยีที่ใช้ (เช่น ภาษา สภาพแวดล้อม เครื่องมือที่เลือกใช้)
- ระบุข้อมูลที่สำคัญที่ต้องรวบรวมสำหรับการพัฒนา เช่น ซอสซอสโค้ด ไฟล์แพ็กเกจ ไลบรารี เอกสารประกอบการจัดเก็บข้อมูล เป็นต้น และระยะเวลาที่ต้องเก็บรักษาไว้ การสิ้นสุดอายุการใช้งานของซอฟต์แวร์
- ตรวจสอบให้แน่ใจว่านโยบายครอบคลุมวงจรชีวิตซอฟต์แวร์ทั้งหมด รวมถึงการแจ้งให้ผู้ใช้ทราบถึงการสิ้นสุดการสนับสนุนซอฟต์แวร์
- ตรวจสอบข้อกำหนดด้านความปลอดภัยทั้งหมด อย่างน้อยปีละ 1 ครั้ง หรือหากมีข้อกำหนดด้านความปลอดภัยใหม่ หรือเหตุการณ์ด้านความปลอดภัยที่สร้างความเสียหายที่อาจส่งผลกระทบกับระบบสารสนเทศ
- ปฏิบัติตามกระบวนการในการจัดการคำขอข้อยกเว้นตามข้อกำหนด รวมถึงการตรวจสอบข้อยกเว้นที่ได้รับอนุมัติทั้งหมดเป็นระยะๆ

4.1.1.2 การกำหนดบทบาทหน้าที่และความรับผิดชอบของการพัฒนาซอฟต์แวร์ (PO.2)

นโยบาย

- เพื่อกำหนดบทบาทหน้าที่และความรับผิดชอบให้ครอบคลุมทุกส่วนของวงจรการพัฒนาซอฟต์แวร์ (SDLC) รวมทั้งให้มีการตรวจสอบบทบาทและความรับผิดชอบที่กำหนดไว้เป็นระยะ ๆ พร้อมทั้งสื่อสารให้ผู้เกี่ยวข้องทราบ
- เพื่อให้มีการฝึกอบรมตามบทบาทหน้าที่สำหรับบุคลากรที่มีความรับผิดชอบในการสนับสนุนการพัฒนา พร้อมทั้งตรวจสอบทักษะความสามารถของบุคลากรการฝึกอบรมตามบทบาทเป็นระยะ

แนวปฏิบัติ

- กำหนดบทบาทและความรับผิดชอบที่เกี่ยวข้องกับกระบวนการวงจรการพัฒนาซอฟต์แวร์ (SDLC) สำหรับนักพัฒนาซอฟต์แวร์ทุกคน
- กำหนดบทบาทหน้าที่และความรับผิดชอบสำหรับเจ้าหน้าที่ด้านความปลอดภัย หัวหน้าโครงการ ผู้บริหารระดับสูง นักพัฒนาซอฟต์แวร์ นักทดสอบซอฟต์แวร์ วิศวกรฝ่ายปฏิบัติการ และบุคคลอื่นที่เกี่ยวข้องในวงจรการพัฒนาซอฟต์แวร์ (SDLC)
- ดำเนินการทบทวนบทบาท หน้าที่ และความรับผิดชอบอย่างน้อยปีละครั้ง
- นำเครื่องมือและกระบวนการไปใช้งานเพื่อส่งเสริมการมีส่วนร่วมระหว่างการพัฒนาซอฟต์แวร์ (SDLC) เช่น การสร้างช่องทางสำหรับพูดคุย หารือของทีมผู้พัฒนา และผู้ที่เกี่ยวข้องอื่น ๆ
- มอบหมายผู้รับผิดชอบเกี่ยวกับซอสโค้ดแต่ละโครงการให้ชัดเจน
- บันทึกผลการฝึกอบรมสำหรับแต่ละบทบาทหน้าที่
- กำหนดประเภทของการฝึกอบรมหรือหลักสูตรที่จำเป็น และสร้างแผนการฝึกอบรมสำหรับแต่ละบทบาทหน้าที่เพื่อให้บรรลุวัตถุประสงค์ที่ต้องการสำหรับแต่ละบทบาทหน้าที่
- พิจารณากำหนดมาตรการในการวัดผลการฝึกอบรมเพื่อพัฒนาเนื้อหาในการอบรม และเพิ่มทักษะความสามารถของผู้ที่เกี่ยวข้อง
- แต่งตั้งเจ้าหน้าที่และมอบหมายผู้รับผิดชอบกระบวนการนำซอฟต์แวร์ขึ้นสู่ระบบงานจริง (Production)
- อบรมบุคลากรทุกคนที่มีบทบาทและความรับผิดชอบที่เกี่ยวข้องกับการพัฒนาเพื่อเสริมสร้างความปลอดภัยในการพัฒนา และความสำคัญของการพัฒนาอย่างปลอดภัย

4.1.1.3 ประยุกต์ใช้เครื่องมือสนับสนุน (PO.3)

นโยบาย

- เพื่อกำหนดชนิดเครื่องมือที่นำมาใช้ รวมถึงแนวทางในการใช้เครื่องมือสำหรับเชื่อมโยงหรือบูรณาการเข้าด้วยกัน

- เพื่อปฏิบัติตามแนวทางการรักษาความปลอดภัย ในการปรับใช้ และบำรุงรักษาเครื่องมือให้เหมาะสม
- เพื่อกำหนดให้มีการตั้งค่าเครื่องมือ สำหรับการสนับสนุนแนวทางการพัฒนาซอฟต์แวร์ที่ปลอดภัย

<u>แนวปฏิบัติ</u>

- ระบุเครื่องมือที่ต้องใช้ตามลักษณะงาน (เช่น การพัฒนา การทดสอบ การตรวจสอบความปลอดภัย) และระบุเครื่องมือที่จำเป็นในแต่ละกลุ่ม
- ระบุเครื่องมือความปลอดภัยที่จะรวมเข้ากับเครื่องมือสำหรับนักพัฒนา เช่น Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST) หรือ Software Composition Analysis (SCA)
- กำหนดชนิดข้อมูลที่ต้องส่งระหว่างเครื่องมือและรูปแบบข้อมูลที่จะนำมาใช้ เช่น (Log Files, Metadata) และรูปแบบ (เช่น JSON, XML) ที่ต้องใช้ในการสื่อสารระหว่างเครื่องมือ
- ตรวจสอบในชุดเครื่องมือว่าเครื่องมือสามารถสร้างหลักฐานที่ไม่สามารถแก้ไขได้เพื่อเพิ่มความน่าเชื่อถือของกระบวนการ
- ใช้เทคโนโลยีอัตโนมัติสำหรับการจัดการและการทำงานร่วมกันของชุดเครื่องมือ เช่น CI/CD Pipelines หรือ Toolchain Orchestration Platforms เพื่อเพิ่มประสิทธิภาพในการจัดการชุดเครื่องมือ
- ประเมิน เลือก และจัดหาเครื่องมือ รวมถึงประเมินความปลอดภัยของแต่ละเครื่องมือที่เหมาะสมและปลอดภัยสำหรับกระบวนการพัฒนาซอฟต์แวร์
- บูรณาการเครื่องมือเข้ากับเครื่องมืออื่น ๆ และกระบวนการการพัฒนาซอฟต์แวร์ที่มีอยู่
- ใช้เครื่องมือรูปแบบใหม่สำหรับการพัฒนา (เช่น pipelines-as-code, toolchains-as-code) เพื่อการทำงานอัตโนมัติและการควบคุมที่ดียิ่งขึ้น
- นำเทคโนโลยีและกระบวนการที่จำเป็นสำหรับการสร้าง (Builds) ที่สามารถทำซ้ำได้ เพื่อช่วยให้การสร้างซอฟต์แวร์ได้ผลลัพธ์ที่เหมือนเดิมทุกครั้ง
- อัปเดต อัพเกรด หรือเปลี่ยนเครื่องมือตามต้องการเพื่อแก้ไขช่องโหว่ของเครื่องมือหรือเพิ่มความสามารถใหม่ๆ ของเครื่องมือ เพื่อจัดการวงจรชีวิตของเครื่องมือให้ทันสมัยและปลอดภัย
- ตรวจสอบเครื่องมือและบันทึกการทำงานของเครื่องมืออย่างต่อเนื่อง เพื่อตรวจหาปัญหาด้านการปฏิบัติงานและความปลอดภัย รวมถึงการละเมิด

นโยบายและพฤติกรรมที่ผิดปกติ เพื่อความมั่นใจในความปลอดภัยและประสิทธิภาพของเครื่องมือ

- ตรวจสอบความสมบูรณ์และที่มาของเครื่องมือเป็นประจำเพื่อตรวจสอบปัญหาที่อาจเกิดขึ้น และป้องกันความเสี่ยงจากแหล่งที่ไม่น่าเชื่อถือ
- กำหนดความถี่ในการตรวจสอบข้อมูลที่เก็บรวบรวม โดยระบุรอบเวลาที่เหมาะสม เช่น รายเดือนหรือรายไตรมาส และสร้างกระบวนการสำหรับการตรวจสอบข้อมูลอย่างมีประสิทธิภาพ
- กำหนดและบังคับใช้นโยบายด้านความปลอดภัยและการเก็บรักษาข้อมูลสำหรับการจัดเก็บและลบข้อมูล เช่น ระยะเวลาการเก็บรักษาและการเข้ารหัสข้อมูล
- กำหนดผู้รับผิดชอบ เช่น นักพัฒนา ผู้จัดการโครงการ ในการสร้างเอกสารหรือข้อมูลที่จำเป็นหากเครื่องมืออัตโนมัติไม่สามารถจัดการได้

4.1.1.4 กำหนด และบังคับใช้เกณฑ์การตรวจสอบด้านความมั่นคงปลอดภัยของซอฟต์แวร์ (PO.4)

<u>นโยบาย</u>

- เพื่อกำหนดเกณฑ์สำหรับการตรวจสอบความปลอดภัยของซอฟต์แวร์ และติดตามตลอดวงจรการพัฒนาซอฟต์แวร์ (SDLC)
- เพื่อนำกระบวนการ และกลไกในการตรวจสอบมาใช้ เพื่อปกป้องข้อมูลที่จำเป็นในการตรวจสอบ

<u>แนวปฏิบัติ</u>

- ตรวจสอบให้แน่ใจว่าเกณฑ์ที่กำหนดสามารถบ่งชี้ได้อย่างเพียงพอว่าความเสี่ยงด้านความปลอดภัยได้รับการจัดการอย่างมีประสิทธิภาพ เช่น กำหนดเกณฑ์ที่ชัดเจนและวัดผลได้สำหรับการประเมินการจัดการความเสี่ยง
- กำหนดตัวชี้วัดผลการดำเนินงานหลัก (KPIs), ตัวชี้วัดความเสี่ยงหลัก (KRIs), คะแนนความรุนแรงของช่องโหว่ และมาตรการอื่น ๆ สำหรับความปลอดภัยของซอฟต์แวร์ เช่น KPI: เวลาเฉลี่ยในการแก้ไขช่องโหว่ (MTTR), KRI: จำนวนช่องโหว่ที่ไม่ได้แก้ไขในช่วงเวลาที่กำหนด
- เพิ่มเกณฑ์ความปลอดภัยของซอฟต์แวร์ในกระบวนการตรวจสอบที่มีอยู่ หรือระบุข้อกำหนดด้านความปลอดภัยให้เป็นส่วนหนึ่งของเงื่อนไขการส่งมอบงาน

- ตรวจสอบซอฟต์แวร์ที่พัฒนาขึ้นเพื่อพิจารณาว่าซอฟต์แวร์ที่พัฒนาตรงตามเกณฑ์หรือไม่ หรือตรวจสอบจากรายงานผลการสแกนความปลอดภัยและการอนุมัติในระบบ
- บันทึกกิจกรรมการอนุมัติ การปฏิเสธ และคำร้องขอยกเว้นด้านความปลอดภัยเพื่อเก็บประวัติการตรวจสอบความปลอดภัยในระบบเพื่อติดตามและตรวจสอบย้อนหลัง
- วิเคราะห์ข้อมูลที่รวบรวมในบริบทของความสำเร็จและความล้มเหลวด้านความปลอดภัยของโครงการพัฒนาแต่ละโครงการ เช่น การตรวจสอบสาเหตุของปัญหา (Root Cause Analysis) เพื่อปรับปรุงกระบวนการพัฒนา SSDLC
- ใช้ชุดเครื่องมือเพื่อรวบรวมข้อมูลโดยอัตโนมัติสำหรับสนับสนุนการตัดสินใจด้านความปลอดภัย เช่น CI/CD Pipelines สำหรับการรวบรวมข้อมูลความปลอดภัย
- ปรับใช้เครื่องมือเพิ่มเติมหากจำเป็น เพื่อสนับสนุนการสร้างและการรวบรวมข้อมูลให้สอดรับเกณฑ์ที่กำหนด เช่น เพิ่มเครื่องมือสแกนช่องโหว่ หรือระบบตรวจสอบกิจกรรม
- ทำให้กระบวนการตัดสินใจเป็นอัตโนมัติโดยใช้เกณฑ์ที่กำหนด และทบทวนกระบวนการเป็นระยะ เช่น การอนุมัติการ Deploy เมื่อผ่านการตรวจสอบความปลอดภัย
- อนุญาตให้เฉพาะบุคคลที่ได้รับอนุญาตเข้าถึงข้อมูลที่รวบรวมไว้ และป้องกันการแก้ไขหรือการลบข้อมูล เช่น การควบคุมสิทธิ์การเข้าถึง (Access Control) และการเข้ารหัสข้อมูลเพื่อความปลอดภัย

4.1.1.5 ดำเนินการและบำรุงรักษาสภาพแวดล้อมที่ปลอดภัยสำหรับการพัฒนาซอฟต์แวร์ (PO.5)

<u>นโยบาย</u>

- เพื่อแยกและปกป้องแต่ละสภาพแวดล้อมที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์
- เพื่อรักษาความปลอดภัยและเสริมความแข็งแกร่งให้กับอุปกรณ์ของผู้ที่เกี่ยวข้อง (เช่น อุปกรณ์ของนักออกแบบซอฟต์แวร์ นักพัฒนา ผู้ทดสอบ ผู้สร้าง เป็นต้น) เพื่อดำเนินงานที่เกี่ยวข้องกับการพัฒนาโดยใช้แนวทางการดำเนินการตามความเสี่ยง (Risk-based Approach)

<u>แนวปฏิบัติ</u>

- ใช้การยืนยันตัวตนแบบหลายปัจจัย (MFA) ที่อิงตามความเสี่ยง และการเข้าถึงแบบมีเงื่อนไข (Conditional Access) สำหรับแต่ละสภาพแวดล้อม เช่น บังคับให้ผู้ใช้ทำการยืนยันตัวตนเพิ่มเติมในกรณีที่ตรวจพบความเสี่ยง เช่น การเข้าถึงจากตำแหน่งที่ไม่ปกติ
- ใช้การแบ่งแยกเครือข่าย (Network Segmentation) และการควบคุมการเข้าถึง (Access Control) เพื่อแยกสภาพแวดล้อมออกจากกัน รวมถึงการแยกองค์ประกอบภายในแต่ละสภาพแวดล้อมที่ไม่ใช่สภาพแวดล้อมการใช้งานจริง (Non-Production Environments) เพื่อลดพื้นที่การโจมตีและการเคลื่อนย้ายของผู้โจมตี รวมถึงลดความเสี่ยงจากการเพิ่มสิทธิ์หรือการเข้าถึงที่ไม่ได้รับอนุญาต
- บังคับใช้นโยบายการยืนยันตัวตนและจำกัดการเชื่อมต่อเข้าและออกจากแต่ละสภาพแวดล้อมพัฒนาซอฟต์แวร์อย่างเข้มงวด
- ลดการเข้าถึงโดยตรงต่อระบบ เช่น บริการ Build และตรวจสอบการเข้าถึงทั้งหมดอย่างต่อเนื่อง หรือบันทึกและตรวจสอบทุกความพยายามในการเข้าถึง รวมถึงการใช้สิทธิ์พิเศษ (Privileged Access)
- ตรวจสอบ และตรวจสอบความสัมพันธ์ด้านความเชื่อถือ (Trust Relationships) สำหรับการอนุญาตและการเข้าถึงระหว่างสภาพแวดล้อมและองค์ประกอบภายในแต่ละสภาพแวดล้อมเป็นประจำ เพื่อตรวจหาความเสี่ยงที่อาจเกิดขึ้นจากการตั้งค่าความเชื่อถือระหว่างระบบ
- ตรวจสอบการทำงานและการแจ้งเตือนในทุกองค์ประกอบของสภาพแวดล้อมการพัฒนาอย่างต่อเนื่อง เพื่อตรวจจับ ตอบสนอง และกู้คืนจากเหตุการณ์ทางไซเบอร์ เช่น ใช้ระบบ SIEM (Security Information and Event Management) เพื่อตรวจสอบและตอบสนองต่อภัยคุกคาม
- กำหนดค่าการควบคุมความปลอดภัยและเครื่องมืออื่น ๆ ที่ใช้ในการปกป้องสภาพแวดล้อมสำหรับการจัดเก็บบันทึกกิจกรรม เช่น จัดเก็บข้อมูลหลักฐานเพื่อการตรวจสอบ เช่น Log การเข้าถึงหรือรายงานการตรวจสอบ
- ตรวจสอบซอฟต์แวร์ทั้งหมดในแต่ละสภาพแวดล้อมอย่างต่อเนื่องเพื่อหาช่องโหว่ใหม่ ๆ และตอบสนองต่อช่องโหว่ตามแนวทางที่แนะนำจากเครื่องมือที่ได้ทดสอบ

- กำหนดค่าและดำเนินการมาตรการรักษาความปลอดภัยสำหรับโครงสร้างพื้นฐานของสภาพแวดล้อมโดยอิงตามสถาปัตยกรรม Zero Trust เช่น การตรวจสอบตัวตนและสิทธิ์การเข้าถึงทุกครั้งแม้อยู่ภายในเครือข่ายเดียวกัน
- กำหนดค่าการตั้งค่าความมั่นคงปลอดภัย เพื่อเพิ่มความปลอดภัย (Hardening Guides) เช่น CIS Benchmarks เพื่อตั้งค่าความปลอดภัย
- กำหนดสิทธิ์การใช้งานสำหรับนักพัฒนาให้มีฟังก์ชันการทำงานที่น้อยที่สุดตามความจำเป็น และบังคับใช้หลักการของสิทธิ์ที่น้อยที่สุด (Least Privilege) เช่น การลดสิทธิ์ของผู้ใช้และบริการให้เฉพาะสิ่งที่จำเป็นสำหรับการทำงาน
- ตรวจสอบสถานะความปลอดภัยของการระบบสำหรับพัฒนา รวมถึงการติดตามและตรวจสอบการใช้งานสิทธิ์พิเศษ (Privileged Access) เช่น ใช้ระบบ Endpoint Detection and Response (EDR) หรือ Security Information and Event Management (SIEM)
- กำหนดค่าการควบคุมความปลอดภัยและเครื่องมืออื่น ๆ ที่เกี่ยวข้องกับการรักษาความปลอดภัยและการเพิ่มความแข็งแกร่งให้จุดสิ้นสุดสำหรับการพัฒนาให้สร้างหลักฐานของกิจกรรม เช่น การบันทึก Log การตั้งค่าความปลอดภัย
- จัดเตรียมระบบสำหรับแยกเครือข่ายการพัฒนาออกจาก Production เพื่อป้องกันความเสี่ยง

4.1.2. การปกป้องซอฟต์แวร์ (PS)

4.1.2.1 ปกป้องซอสโค้ดทุกรูปแบบจากการเข้าถึงและการดัดแปลงโดยไม่ได้รับอนุญาต (PS.1)

นโยบาย

- เพื่อจัดเก็บข้อมูลซอสโค้ดทุกรูปแบบ (เช่น Source code, Executable code, Configuration-as-code) โดยยึดตามหลักการของการให้สิทธิขั้นต่ำ (Least privilege) โดยกำหนดให้เฉพาะบุคลากร เครื่องมือ และบริการที่ได้รับอนุญาตเท่านั้นในการเข้าถึงได้

แนวปฏิบัติ

- จัดเก็บซอสซอสโค้ดต้นฉบับ และการกำหนดค่าผ่านซอสซอสโค้ด (Configuration-as-Code) และจำกัดการเข้าถึงตามลักษณะของซอสโค้ด เช่น ซอสโค้ดโอเพนซอร์สที่สามารถเข้าถึงได้ต้องได้รับการปกป้องความสมบูรณ์และความพร้อมใช้งาน รวมถึงซอสโค้ดอื่นๆต้องได้รับการปกป้องรักษาความลับด้วยเช่นกัน

- ใช้การควบคุมเวอร์ชันของซอสโค้ดเพื่อติดตามการเปลี่ยนแปลงทั้งหมดที่เกิดขึ้นกับซอสโค้ด และให้สามารถตรวจสอบการเปลี่ยนแปลงได้
- ให้เจ้าของซอสโค้ดตรวจสอบและอนุมัติการเปลี่ยนแปลงที่ดำเนินการโดยผู้อื่นก่อนนำไปใช้งาน
- ใช้การลงนามซอสโค้ด (Code Signing) เพื่อช่วยปกป้องความถูกต้องของไฟล์ที่สามารถรันได้ (Executables)
- ใช้วิทยาการเข้ารหัสลับ เพื่อช่วยตรวจสอบความถูกต้องและสมบูรณ์ของซอสโค้ด (เช่น การ hash code)

4.1.2.2 จัดทำกลไกเพื่อตรวจสอบความสมบูรณ์ และถูกต้องของการเผยแพร่ซอฟต์แวร์ (PS.2)

นโยบาย

- ทำให้การตรวจสอบความสมบูรณ์ของซอฟต์แวร์ ให้ผู้รับซอฟต์แวร์สามารถเข้าถึงได้อย่างปลอดภัยและชัดเจน

แนวปฏิบัติ

- เผยแพร่ไฟล์แฮช (Cryptographic Hashes) ของซอฟต์แวร์ที่ปล่อยออกมา (Release Files) บนเว็บไซต์ที่มีความปลอดภัยสูง
- ใช้หน่วยงานออกใบรับรอง (Certificate Authority) ที่เป็นที่ยอมรับสำหรับการลงนามซอสโค้ด (Code Signing) เพื่อให้ระบบปฏิบัติการ เครื่องมือ หรือบริการของผู้ใช้สามารถตรวจสอบความถูกต้องของลายเซ็นก่อนใช้งานได้
- ทบทวนกระบวนการลงนามซอสโค้ดเป็นระยะ ๆ รวมถึงการต่ออายุ การหมุนเวียน การเพิกถอน และการป้องกันใบรับรอง

4.1.2.3 การจัดเก็บ (Archive) และปกป้อง (Protect) ซอฟต์แวร์แต่ละเวอร์ชันที่ได้เผยแพร่ (PS.3)

นโยบาย

- จัดเก็บไฟล์และข้อมูลสนับสนุนที่จำเป็นอย่างปลอดภัย (เช่น ข้อมูลการตรวจสอบความสมบูรณ์ของซอฟต์แวร์ และข้อมูลแหล่งที่มา) เพื่อให้สามารถเก็บรักษาไว้สำหรับแต่ละรุ่นของซอฟต์แวร์
- รวบรวม ปกป้อง ดูแลรักษา และแบ่งปันข้อมูลแหล่งที่มาของซอสโค้ดในแต่ละส่วนประกอบทั้งหมดของแต่ละรุ่นซอฟต์แวร์ (เช่น ในรายการวัตถุดิบซอฟต์แวร์ หรือ Software Bill of Materials [SBOM])

แนวปฏิบัติ

- จัดเก็บไฟล์ซอฟต์แวร์สำหรับเผยแพร่ (Release Files) รูปภาพที่เกี่ยวข้อง และอื่น ๆ ไว้ในที่เก็บข้อมูล (Repository) ตามนโยบายที่องค์กรกำหนด โดยอนุญาตให้เฉพาะบุคลากรที่จำเป็นสามารถเข้าถึงได้ในโหมดอ่านอย่างเดียว (Read-Only) และไม่อนุญาตให้บุคคลอื่นเข้าถึง

- จัดเก็บและปกป้องข้อมูลการตรวจสอบความสมบูรณ์ของรุ่นซอฟต์แวร์ (Release Integrity Verification Information) และข้อมูลแหล่งที่มา (Provenance Data) เช่น โดยการแยกเก็บไว้ในตำแหน่งที่แตกต่างจากซอฟต์แวร์ หรือโดยการลงนามข้อมูลดังกล่าว (Signing the Data)
- ทำให้ข้อมูลแหล่งที่มา (Provenance Data) พร้อมใช้งานสำหรับผู้ที่ได้รับซอฟต์แวร์ ตามนโยบายขององค์กร โดยใช้รูปแบบที่เป็นมาตรฐาน
- ทำให้ข้อมูลแหล่งที่มาพร้อมใช้งานสำหรับทีมปฏิบัติการและทีมรับมือขององค์กร เพื่อช่วยในการลดความเสี่ยงจากช่องโหว่ของซอฟต์แวร์
- จัดเตรียมวิธีการให้ผู้รับสามารถตรวจสอบความถูกต้องของข้อมูลแหล่งที่มาได้

4.1.3. ผลิตซอฟต์แวร์ที่มีความปลอดภัย (PW)

4.1.3.1 การพัฒนาซอฟต์แวร์ให้เป็นไปตามความต้องการด้านความมั่นคงปลอดภัยเพื่อลดความเสี่ยง (PW.1)

นโยบาย

- สร้างแบบจำลองความเสี่ยง เช่น การสร้างแบบจำลองภัยคุกคาม การสร้างแบบจำลองการโจมตี เพื่อช่วยประเมินความเสี่ยงด้านความปลอดภัยสำหรับซอฟต์แวร์
- ติดตามข้อกำหนดด้านความมั่นคงปลอดภัยสารสนเทศ ความเสี่ยง ที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์

แนวปฏิบัติ

- ฝึกอบรมทีมพัฒนา (โดยเฉพาะผู้เชี่ยวชาญด้านความปลอดภัย) หรือร่วมมือกับผู้เชี่ยวชาญเพื่อสร้างแบบจำลองและวิเคราะห์วิธีการแก้ไขความเสี่ยงเพื่อสื่อสารความเสี่ยงและกำหนดวิธีการแก้ไข รวมถึงการนำมาตรการบรรเทาความเสี่ยงมาใช้
- ติดตามความเสี่ยงและสถิติสำหรับซอฟต์แวร์ก่อนหน้าเพื่อแจ้งการประเมินความเสี่ยงด้านความปลอดภัย
- ใช้หลักการจำแนกประเภทข้อมูลเพื่อระบุและกำหนดลักษณะข้อมูลแต่ละประเภท
- บันทึกการตอบสนองต่อความเสี่ยงแต่ละประเภท รวมถึงวิธีการลดความเสี่ยงและเหตุผลในยอมรับความเสี่ยง
- ประเมินข้อยกเว้นที่ได้รับการอนุมัติทั้งหมดตามข้อกำหนดด้านความปลอดภัยเป็นระยะๆ
- กำหนดค่าความปลอดภัยสำหรับโมดูลเพื่อรองรับคุณสมบัติและบริการด้านความปลอดภัยตามมาตรฐาน และทำให้การกำหนดค่าพร้อมใช้งาน เพื่อให้นักพัฒนาสามารถใช้งานได้ทันที

- กำหนดเกณฑ์ที่ซอฟต์แวร์จะต้องรองรับคุณลักษณะและบริการด้านความปลอดภัยที่จะพัฒนา

4.1.3.2 ตรวจสอบการออกแบบซอฟต์แวร์เพื่อยืนยันว่าเป็นไปตามข้อกำหนดด้านความมั่นคงปลอดภัย และข้อมูลความเสี่ยง (PW.2)

นโยบาย

- กำหนดให้บุคคลที่มีคุณสมบัติเหมาะสม (หรือคณะบุคคล) ที่ไม่ได้ มีส่วนเกี่ยวข้องกับการออกแบบ ตรวจสอบการออกแบบซอฟต์แวร์เพื่อยืนยันว่าซอฟต์แวร์นั้นเป็นไปตามข้อกำหนดด้านความปลอดภัย

แนวปฏิบัติ

- ตรวจสอบการออกแบบซอฟต์แวร์เพื่อยืนยันว่าเป็นไปตามข้อกำหนดด้านความปลอดภัย
- ตรวจสอบความเสี่ยงระหว่างการออกแบบซอฟต์แวร์ เพื่อพิจารณาว่าพบความเสี่ยงที่อาจส่งผลกระทบหรือไม่
- ให้ผู้ออกแบบซอฟต์แวร์แก้ไขข้อผิดพลาดเพื่อให้เป็นไปตามข้อกำหนด
- เปลี่ยนแปลงการออกแบบและ/หรือกลยุทธ์การตอบสนองความเสี่ยงหากไม่สามารถปฏิบัติตามข้อกำหนดด้านความปลอดภัยได้

4.1.3.3 การนำซอฟต์แวร์เดิมที่มีความปลอดภัยกลับมาใช้งานใหม่ โดยไม่ใช้วิธีการทำซ้ำ (Duplication) (PW.3)

นโยบาย

- จัดหาและบำรุงรักษาส่วนประกอบซอฟต์แวร์ที่ปลอดภัย (เช่น ไลบรารีซอฟต์แวร์ โมดูล มิดเดิลแวร์ เฟรมเวิร์ก)
- ตรวจสอบส่วนประกอบซอฟต์แวร์เชิงพาณิชย์ โอเพนซอร์ส ที่มีการใช้ให้เป็นไปตามข้อกำหนดตามที่องค์กรกำหนดตลอดช่วงของวงจรการพัฒนาซอฟต์แวร์ (SDLC)

แนวปฏิบัติ

- กำหนดค่าการรักษาความปลอดภัยสำหรับส่วนประกอบซอฟต์แวร์
- จัดทำรายชื่อส่วนประกอบซอฟต์แวร์และเวอร์ชันส่วนประกอบของซอฟต์แวร์
- ปฏิบัติตามแนวทางการรักษาความปลอดภัยที่องค์กรกำหนดไว้ สำหรับการพัฒนาซอฟต์แวร์ที่ปลอดภัย
- บำรุงรักษาที่เก็บซอฟต์แวร์ สำหรับส่วนประกอบของซอฟต์แวร์
- ตรวจสอบเป็นประจำว่ามีช่องโหว่ที่เปิดเผยต่อสาธารณะ และบริการที่ผู้จำหน่ายยังไม่ได้แก้ไขหรือไม่
- สร้างการตรวจจับอัตโนมัติของช่องโหว่ที่ทราบในส่วนประกอบซอฟต์แวร์

- ตรวจสอบให้แน่ใจว่าส่วนประกอบซอฟต์แวร์แต่ละส่วนได้รับการบำรุงรักษาอย่างต่อเนื่องและยังไม่ถึงจุดสิ้นสุดอายุการใช้งานทั้งนี้ควรรวมถึงช่องโหว่ใหม่ๆ ที่พบในซอฟต์แวร์ที่กำลังได้รับการแก้ไข
- กำหนดแผนปฏิบัติการสำหรับซอฟต์แวร์แต่ละส่วนที่ไม่ได้รับการบำรุงรักษาอีกต่อไปหรือจะไม่สามารถใช้งานได้ในอนาคตอันใกล้นี้

4.1.3.4 การเขียนซอสโค้ดโดยยึดหลักการพัฒนาระบบอย่างปลอดภัย (Secure coding) (PW.4)

นโยบาย

- ปฏิบัติตามแนวทางการเขียนซอสโค้ดที่ปลอดภัยทั้งหมดที่เหมาะสมกับภาษาและสภาพแวดล้อมการพัฒนาเพื่อตอบสนองข้อกำหนดขององค์กร

แนวปฏิบัติ

- ตรวจสอบอินพุตทั้งหมดและตรวจสอบและเข้ารหัสเอาต์พุตทั้งหมดอย่างถูกต้อง
- หลีกเลี่ยงการใช้ ฟังก์ชันที่ไม่ปลอดภัย
- ตรวจจับข้อผิดพลาดและจัดการอย่างเหมาะสม
- ให้มีความสามารถในการบันทึกและการติดตาม
- ตรวจสอบช่องโหว่อื่นๆ ที่พบได้ ทั่วไปในภาษาและสภาพแวดล้อมการพัฒนา

4.1.3.5 การกำหนดกระบวนการในการพัฒนาความปลอดภัย และกำหนดการตั้งค่าเครื่องมือคอมไพเลอร์ และอินเทอร์พรีเตอร์ที่องค์กรใช้งาน (PW.5)

นโยบาย

- ใช้เครื่องมือ คอมไพเลอร์ อินเทอร์พรีเตอร์ และเครื่องมืออื่น ๆ ที่มีความสามารถในการเพิ่มความปลอดภัยของการพัฒนาซอฟต์แวร์
- กำหนดว่าควรใช้ฟีเจอร์คอมไพเลอร์ อินเทอร์พรีเตอร์ และเครื่องมือใด ๆ โดยกำหนดค่าแต่ละฟีเจอร์อย่างปลอดภัยตามที่ได้รับอนุญาติ ก่อนนำไปใช้งาน

แนวปฏิบัติ

- ใช้คอมไพเลอร์อินเตอร์พรีเตอร์ และเครื่องมือบิลด์เวอร์ชันล่าสุด
- ปฏิบัติตามกระบวนการจัดการการเปลี่ยนแปลงเมื่อปรับใช้ หรืออัปเดตคอมไพเลอร์อินเทอร์พรีเตอร์ และเครื่องมือสร้าง และตรวจสอบการเปลี่ยนแปลงที่ไม่คาดคิดทั้งหมดในเครื่องมือ
- เปิดใช้งานคุณลักษณะคอมไพเลอร์ที่สามารถแจ้งเตือนสำหรับซอสโค้ดที่ไม่ปลอดภัย ในระหว่างกระบวนการคอมไพเลอร์
- ทดสอบเพื่อให้แน่ใจว่าคุณสมบัติต่างๆ ทำงานตามที่คาดหวังและไม่ทำให้ เกิดปัญหาการทำงานหรือปัญหาอื่น ๆ โดยไม่ได้ตั้งใจ
- ตรวจสอบอย่างต่อเนื่องว่ามีการใช้ การกำหนดค่าที่ได้รับอนุมัติหรือไม่

4.1.3.6 การวิเคราะห์และทบทวนซอสโค้ดที่สามารถอ่านได้โดยมนุษย์เพื่อระบุช่องโหว่ และยืนยันว่าเป็นไปตามข้อกำหนดด้านความมั่นคงปลอดภัย (PW.6)

นโยบาย

- ควรมีการตรวจสอบซอสโค้ด และการวิเคราะห์ ซอสโค้ด ตามที่องค์กรกำหนดไว้
- ดำเนินการตรวจสอบซอสโค้ด และวิเคราะห์ซอสโค้ดตามมาตรฐาน

แนวปฏิบัติ

- ปฏิบัติตามนโยบายหรือแนวทางขององค์กรเกี่ยวกับเวลาที่ควรดำเนินการตรวจสอบซอสโค้ดและวิธีการดำเนินการ ซึ่งอาจรวมถึงซอสโค้ดของบุคคลที่สาม และโมดูลซอสโค้ดที่นำมาใช้ซ้ำ
- เลือกวิธีการตรวจสอบและ/หรือวิเคราะห์ซอสโค้ดตามขั้นตอนของซอฟต์แวร์
- ใช้ผู้ตรวจสอบผู้เชี่ยวชาญเพื่อตรวจสอบซอสโค้ดว่ามีแบ็คดอร์หรือเนื้อหาที่เป็นอันตรายอื่นๆ หรือไม่
- ใช้เครื่องมือตรวจสอบ เพื่อช่วยอำนวยความสะดวกในกระบวนการตรวจสอบ
- ใช้เครื่องมือวิเคราะห์เพื่อตรวจสอบซอสโค้ดโดยอัตโนมัติว่ามีช่องโหว่ และความสอดคล้องกับมาตรฐานการหรือไม่
- บันทึกบทเรียนที่เรียนรู้จากการตรวจสอบและวิเคราะห์ซอสโค้ดในวิธีที่นักพัฒนาสามารถเข้าถึงและค้นหาได้

4.1.3.7 การทดสอบ Executable code เพื่อระบุช่องโหว่ และยืนยันว่าเป็นไปตามข้อกำหนดด้านความมั่นคงปลอดภัย (PW.7)

นโยบาย

- กำหนดให้ดำเนินการวิเคราะห์ และทดสอบ Executable code เพื่อค้นหาช่องโหว่ที่ไม่ได้ระบุไว้ในการตรวจสอบ
- กำหนดขอบเขต ออกแบบ ดำเนินการ และบันทึกผลการทดสอบ รวมถึงบันทึกและจำแนกช่องโหว่ทั้งหมดที่พบ พร้อมทั้งระบุแนวทางแก้ไขให้แก่ทีมพัฒนา

แนวปฏิบัติ

- กำหนดนโยบายหรือแนวปฏิบัติขององค์กรเกี่ยวกับการตรวจสอบซอสโค้ดโดยผู้ที่ได้รับมอบหมาย (Code review) และการตรวจสอบโดยเครื่องมืออัตโนมัติ (Code analysis) อย่างสม่ำเสมอ โดยอาจรวมถึงซอสโค้ดจากภายนอกและโมดูลซอสโค้ดที่เขียนขึ้นภายในองค์กร
- ผู้ที่เกี่ยวข้องสามารถเลือกใช้วิธีการในการตรวจสอบซอสโค้ดโดยพิจารณาตามความเหมาะสมของขั้นตอน และสถานะการพัฒนาซอฟต์แวร์นั้น ๆ

- ผลสานการทดสอบช่องโหว่แบบไดนามิก (เช่น การทดสอบในขณะที่ระบบกำลังใช้งาน) เพื่อบันทึกและจัดการปัญหาที่ค้นพบทั้งหมด รวมถึงชุดการทดสอบอย่างเป็นระบบ
- ให้พิจารณานำการทดสอบเจาะระบบเพื่อจำลองสถานการณ์ที่มีความเสี่ยงสูงในการโจมตีซอฟต์แวร์
- ระบุและบันทึกสาเหตุของปัญหาที่ค้นพบ
- บันทึกบทเรียนที่ได้เรียนรู้จากการทดสอบซอสโค้ดไว้ในระบบบันทึกที่นักพัฒนาสามารถเข้าเข้าถึงและค้นหาได้

4.1.3.8 การตั้งค่าซอฟต์แวร์ให้ปลอดภัยตามค่าเริ่มต้น (PW.8)

<u>นโยบาย</u>

- กำหนดแนวทางปฏิบัติในการตั้งค่าระบบให้มีความมั่นคงปลอดภัย และเป็นแนวทางในการกำหนดการตั้งค่าเริ่มต้นอย่างปลอดภัย โดยไม่ทำให้ฟังก์ชันความปลอดภัยที่จัดทำโดยแพลตฟอร์ม โครงสร้างพื้นฐานเครือข่าย หรือบริการต่าง ๆ อ่อนแอลง
- บังคับใช้การตั้งค่าเริ่มต้นบนซอฟต์แวร์หรือระบบงานที่เกี่ยวข้อง และกำหนดให้ผู้ดูแลระบบซอฟต์แวร์บันทึกรายการการตั้งค่าเป็นลายลักษณ์อักษร

<u>แนวปฏิบัติ</u>

- ดำเนินการทดสอบเพื่อให้แน่ใจว่าการตั้งค่า รวมถึงการตั้งค่าเริ่มต้น ทำงานตามที่คาดหวัง และไม่ทำให้เกิดจุดอ่อนด้านความปลอดภัย ปัญหาการทำงาน หรือปัญหาอื่น ๆ โดยไม่ได้ตั้งใจ
- ดำเนินการตรวจสอบการตั้งค่าของซอฟต์แวร์ให้เป็นไปตามการตั้งค่าที่ได้รับอนุมัติ
- จัดทำเอกสารรายการเกี่ยวกับวัตถุประสงค์ของการตั้งค่าในแต่ละรายการ เช่น ค่าเริ่มต้น การตั้งค่าด้านการรักษาความปลอดภัย การตั้งค่าที่อาจกระทบต่อการปฏิบัติงาน และการตั้งค่าอื่น ๆ เป็นต้น
- ใช้กลไกทางเทคนิคเพื่อบันทึกพฤติกรรมของผู้ดูแลระบบในการตั้งค่า และประเมินผล เพื่อพิจารณาความถูกต้องเหมาะสม
- จัดเก็บการตั้งค่าเริ่มต้นในรูปแบบที่พร้อมใช้งาน และปฏิบัติตามแนวทางการควบคุมการเปลี่ยนแปลงในการแก้ไข

4.1.4. การตอบสนองต่อช่องโหว่ (RV)

4.1.4.1 การระบุและยืนยันช่องโหว่อย่างต่อเนื่อง (RV.1)

นโยบาย

- รวบรวมข้อมูลจากผู้ใช้งาน และแหล่งข้อมูลสาธารณะเกี่ยวกับช่องโหว่ของส่วนประกอบซอฟต์แวร์ (Software components) จากภายนอกที่องค์กรใช้งาน และพิจารณาตรวจสอบจากแหล่งข้อมูลที่น่าเชื่อถือทั้งหมด
- ตรวจสอบ วิเคราะห์ และทดสอบซอสโค้ดของซอฟต์แวร์เพื่อระบุช่องโหว่ที่เกิดขึ้นในซอฟต์แวร์ขององค์กร
- กำหนดนโยบาย และกระบวนการที่จำเป็นเกี่ยวกับการเปิดเผยและแก้ไขช่องโหว่ พร้อมทั้งกำหนดบทบาท หน้าที่ความรับผิดชอบของเจ้าหน้าที่ที่เกี่ยวข้องกับนโยบาย และกระบวนการดังกล่าว

แนวปฏิบัติ

- ตรวจสอบฐานข้อมูลช่องโหว่ (Vulnerability database) , Security mailing lists และแหล่งรายงานช่องโหว่อื่น ๆ ผ่านวิธีการตรวจติดตามด้วยตนเอง หรือด้วยวิธีการแบบอัตโนมัติ
- ใช้แหล่งข้อมูลข่าวกรองด้านภัยคุกคาม เพื่อทำความเข้าใจแนวทางพื้นฐานในการใช้งานช่องโหว่
- ตรวจสอบที่มาและข้อมูลองค์ประกอบของซอฟต์แวร์ หรือส่วนประกอบของซอฟต์แวร์ (Software components) ทั้งหมดในรูปแบบอัตโนมัติ เพื่อระบุช่องโหว่ใหม่ ๆ ที่เกิดขึ้น
- กำหนดให้มีการตั้งค่าเครื่องมือ (toolchain) ที่ใช้ในการวิเคราะห์และทดสอบซอสโค้ด ให้สามารถดำเนินการได้ตามรอบที่องค์กรกำหนด เพื่อให้ซอฟต์แวร์ที่อยู่ในระหว่างการสนับสนุนได้รับการตรวจติดตามอย่างต่อเนื่องและสม่ำเสมอ
- จัดทำช่องทางการเปิดเผยช่องโหว่เพื่ออำนวยความสะดวกให้นักวิจัยด้านความปลอดภัย และผู้ที่เกี่ยวข้อง สามารถเรียนรู้เกี่ยวกับซอฟต์แวร์ขององค์กร และสามารถรายงานช่องโหว่ที่อาจเกิดขึ้นได้
- กำหนดกระบวนการในการตอบสนองต่อเหตุการณ์ด้านความมั่นคงปลอดภัย รวมถึงกำหนดแผนการสื่อสารต่อผู้มีส่วนได้ส่วนเสียขององค์กรกรณีเกิดเหตุการณ์ด้านความมั่นคงปลอดภัย
- กำหนดให้ดำเนินการฝึกซ้อมกระบวนการในการตอบสนองเหตุการณ์ด้านความมั่งคงปลอดภัยของผลิตภัณฑ์อย่างสม่ำเสมอ

4.1.4.2 การประเมิน จัดลำดับความสำคัญ และจัดการช่องโหว่ที่พบ (RV.2)

นโยบาย

- วิเคราะห์ช่องโหว่ที่พบ และรวบรวมข้อมูลความเสี่ยงที่อาจเกิดขึ้นเพื่อวางแผนการจัดการ แก้ไข และตอบสนองต่อความเสี่ยง
- กำหนดวางแผนและบังคับใช้แนวทางการตอบสนองความเสี่ยงต่อช่องโหว่ต่าง ๆ

แนวปฏิบัติ

- ประยุกต์ใช้ซอฟต์แวร์ติดตามปัญหาที่ใช้งานเพื่อบันทึกช่องโหว่ของระบบ
- ดำเนินการคำนวณความเสี่ยงในแต่ละช่องโหว่โดยอิงจากความยากง่ายในการใช้งานช่องโหว่เพื่อโจมตี ผลกระทบที่อาจเกิดขึ้นหากองค์กรถูกโจมตีโดยช่องโหว่ดังกล่าว และเกณฑ์อื่น ๆ ที่เกี่ยวข้องตามเหมาะสมขององค์กร
- ดำเนินการจัดการช่องโหว่โดยพิจารณาจากความเสี่ยงที่ถูกประเมิน (เช่น การบรรเทาความเสี่ยง การยอมรับความเสี่ยง การถ่ายโอนความเสี่ยง การหลีกเลี่ยงความเสี่ยง) พร้อมทั้งกำหนดลำดับความสำคัญในการจัดการช่องโหว่
- กรณีที่ยังไม่สามารถใช้แนวทางในการบรรเทาความเสี่ยงของช่องโหว่อย่างถาวรได้ ให้พิจารณาใช้แนวทางบรรเทาผลกระทบชั่วคราว (Workarounds) ในระหว่างการพิจารณาแนวทางการบรรเทาอย่างถาวร และเพิ่มแนวทางการแก้ไขชั่วคราว (Workarounds) ดังกล่าวลงในแผนการแก้ไขช่องโหว่
- พัฒนาและเผยแพร่คำแนะนำด้านความปลอดภัย เพื่อให้ข้อมูลที่จำเป็นแก่ผู้ใช้งานซอฟต์แวร์ รวมถึงคำอธิบายเกี่ยวกับช่องโหว่ต่าง ๆ วิธีค้นหาซอฟต์แวร์ที่มีช่องโหว่ และวิธีการแก้ไข (เช่น การปรับปรุงแพตช์ การปรับเปลี่ยนการตั้งค่าของซอฟต์แวร์ วิธีการแก้ไขแบบชั่วคราว)
- ปรับปรุงบันทึกแนวทางการออกแบบซอฟต์แวร์ การตอบสนองต่อความเสี่ยง และข้อยกเว้นที่ได้รับอนุมัติตามความจำเป็น (อ้างอิงข้อ 4.1.3.1 การพัฒนาซอฟต์แวร์ให้เป็นไปตามความต้องการด้านความมั่นคงปลอดภัยเพื่อลดความเสี่ยง)

4.1.4.3 การวิเคราะห์เพื่อหาสาเหตุที่แท้จริงของช่องโหว่ (RV.3)

นโยบาย

- วิเคราะห์ถึงสาเหตุที่แท้จริงของช่องโหว่ที่ตรวจพบ
- ตรวจสอบและวิเคราะห์ความเหมาะสมของสาเหตุที่แท้จริงโดยให้มีรอบในการตรวจสอบความเหมาะสม
- พิจารณาซอฟต์แวร์ที่มีลักษณะ หรือถูกสร้างขึ้นบนพื้นฐานเดียวกันกับซอฟต์แวร์ที่ตรวจพบช่องโหว่ โดยสามารถแก้ไขได้ทันที
- ตรวจสอบกระบวนการของวงจรการพัฒนาซอฟต์แวร์ (SDLC) และพิจารณาปรับปรุงตามความเหมาะสมเพื่อป้องกัน หรือลดโอกาสการเกิดขึ้นของสาเหตุที่แท้จริงในการสร้าง หรือพัฒนาซอฟต์แวร์

แนวปฏิบัติ

- วิเคราะห์และบันทึกสาเหตุที่แท้จริงของปัญหาที่ตรวจพบ
- จัดทำบันทึกเพื่อให้ผู้พัฒนาสามารถค้นหา และเข้าถึงข้อมูลแนวทางการแก้ไขของปัญหาที่เกิดขึ้น พร้อมทั้งแนวทางการป้องกันไม่ให้เกิดซ้ำ
- พิจารณาเพิ่มมาตรการในการตรวจสอบถึงปัญหา และสาเหตุของการเกิดช่องโหว่โดยอัตโนมัติบนเครื่องมือที่ใช้ในกระบวนการของวงจรการพัฒนาซอฟต์แวร์ (SDLC)
- ปรับปรุงแนวทาง และคู่มือในการตรวจสอบสาเหตุของการเกิดช่องโหว่
- กำหนดรอบในการทบทวนกระบวนการของวงจรการพัฒนาซอฟต์แวร์ (SDLC) อย่างสม่ำเสมอ

4.2. นโยบายและแนวปฏิบัติในการพัฒนาระบบให้มีความมั่นคงปลอดภัย ตามมาตรฐาน ISO/IEC 27001

4.2.1. ความต้องการด้านความมั่นคงปลอดภัยของระบบ

นโยบาย

- เพื่อให้ระบบมีความมั่นคงปลอดภัยตลอดวงจรการพัฒนาระบบ ต้องมีการกำหนดความต้องการด้านความมั่นคงปลอดภัยของระบบในการพัฒนาระบบ

แนวปฏิบัติ

- กำหนดความต้องการสำหรับทำให้ระบบมีความมั่นคงปลอดภัยต้องมีการพิจารณาร่วมกับความต้องการสำหรับระบบการพัฒนาระบบงานใหม่ หรือปรับปรุงระบบงานเดิมที่มีอยู่แล้ว
- ระบบงานที่มีบริการส่งข้อมูลสารสนเทศผ่านเครือข่ายสาธารณะต้องได้รับการออกแบบ พัฒนาให้สามารถป้องกันการเปิดเผย และเปลี่ยนแปลงข้อมูลสารสนเทศโดยไม่ได้รับอนุญาตอย่างเหมาะสม
- ข้อมูลสารสนเทศที่เกี่ยวข้องกับการทำธุรกรรมขององค์กร ต้องได้รับการป้องกันการรับส่งข้อมูลที่ไม่สมบูรณ์ การส่งข้อมูลผิดเส้นทาง การเปลี่ยนแปลงข้อความโดยไม่ได้รับอนุญาต การเปิดเผยข้อมูลโดยไม่ได้รับอนุญาต และการส่งข้อมูลซ้ำ

4.2.2. ความมั่นคงปลอดภัยในกระบวนการพัฒนาระบบ

นโยบาย

- เพื่อให้ระบบมีความมั่นคงปลอดภัยตลอดวงจรการพัฒนาระบบ ต้องกำหนดเป็นขั้นปฏิบัติขึ้นมาเพื่อเป็นแนวทางในการพัฒนาระบบ

แนวปฏิบัติ

- ควบคุมการเปลี่ยนแปลงที่เกิดขึ้นระหว่างการพัฒนาระบบ เพื่อลดผลกระทบต่อความมั่นคงปลอดภัยของระบบที่ได้ออกแบบไว้ และผลกระทบต่อเป้าหมายของการพัฒนาระบบด้วย
- ควบคุมการเปลี่ยนแปลงที่เกิดขึ้นระหว่างการพัฒนาระบบ เพื่อลดผลกระทบต่อความมั่นคงปลอดภัยของระบบที่ได้ออกแบบไว้ และผลกระทบต่อเป้าหมายของการพัฒนาระบบด้วย
- กรณีมีการเปลี่ยนแปลงโครงสร้างพื้นฐานของระบบงาน ต้องมีการทบทวนและทดสอบความมั่นคงปลอดภัยของระบบงานนั้น เพื่อให้มั่นใจว่าไม่มีผลกระทบในทางลบต่อการดำเนินงานขององค์กร
- ต้องพิจารณาและกำหนดสภาพแวดล้อมที่เหมาะสมสำหรับใช้ในการพัฒนาระบบงานที่มีความมั่นคงปลอดภัย เช่น เป็นพื้นที่ควบคุมการเข้าถึงได้ ไม่อยู่ในบริเวณที่มีคนพลุกพล่าน
- ในการพัฒนาระบบสารสนเทศต้องแยกระบบสำหรับการพัฒนาออกจากระบบอื่น ๆ โดยต้องมั่นใจว่าสภาพแวดล้อมของการพัฒนาระบบสารสนเทศ มีความมั่นคงปลอดภัยเพียงพอ
- การจ้างพัฒนาระบบงาน ต้องมีการควบคุม เฝ้าระวัง ติดตามการดำเนินงานอย่างใกล้ชิด เพื่อให้ผลลัพธ์เป็นไปตามขอบเขตงานที่กำหนดตรงตามความต้องการ
- ระบบงานใหม่ ระบบงานที่ปรับปรุง และระบบงานเวอร์ชันใหม่ต้องได้รับทดสอบการใช้งานและการทดสอบคุณสมบัติด้านความมั่นคงปลอดภัย
- ต้องควบคุม และกำหนดสิทธิการเข้าถึง Source Code ของระบบงานอย่างเหมาะสมให้สอดคล้องตามบทบาทหน้าที่ความรับผิดชอบตามที่ได้รับมอบหมาย
- ต้องมีการตรวจทานความถูกต้องของข้อมูลที่นำเข้าระบบงาน เพื่อลดความผิดพลาดในการประมวลผลข้อมูล
- ควรมีการควบคุม และตรวจสอบการทำงานของระบบงาน เพื่อป้องกันความเสี่ยง ที่อาจเกิด จากการทำงาน หรือการประมวลผลข้อมูลที่ผิดพลาดอันจะส่งผลเสียต่อระบบงานโดยรวม
- ต้องกำหนดระบุและอนุมัติข้อกำหนดด้านการรักษาความปลอดภัยของระบบสารสนเทศเมื่อต้องพัฒนา หรือเมื่อมีการนำแอปพลิเคชันมาใช้งาน

- ปฏิบัติตามหลักการเขียนซอสโค้ดอย่างมั่นคงปลอดภัย พร้อมทั้งให้คำแนะนำและช่วยกันดูแลให้มีการเขียนซอสโค้ดอย่างมั่นคงปลอดภัยครอบคลุมทั้งซอฟต์แวร์จากบุคคลภายนอกและซอฟต์แวร์แบบโอเพ่นซอร์ส เพื่อช่วยลดช่องโหว่ด้านความมั่นคงปลอดภัยที่อาจเกิดขึ้นในซอฟต์แวร์

4.2.3. การควบคุมข้อมูลในการทดสอบ

นโยบาย

- เพื่อให้ระบบมีความมั่นคงปลอดภัยตลอดวงจรการพัฒนาระบบ ต้องกำหนดแนวปฏิบัติในการควบคุมข้อมูลในการทดสอบ เพื่อเป็นแนวทางในการพัฒนาระบบให้มีความปลอดภัยต่อข้อมูลที่ใช้ในการทดสอบ

แนวปฏิบัติ

- ต้องหลีกเลี่ยงการใช้ข้อมูลจริงที่มีอยู่บนระบบงานให้บริการมาใช้ในการทดสอบ กรณีจำเป็นที่ต้องใช้ข้อมูลสำเนาของข้อมูลจริงไปทดสอบระบบงานต้องมีการควบคุมข้อมูลที่ใช้ทดสอบอย่างมั่นคงปลอดภัย
- หากข้อมูลที่นำใช้ไปเป็นข้อมูลส่วนบุคคลจะต้องมีการนำไปใช้อย่างไม่ขัดต่อการใช้ข้อมูลส่วนบุคคล

4.3. นโยบายและแนวปฏิบัติด้านการจัดการความมั่นคงปลอดภัยของข้อมูลส่วนบุคคล ตามมาตรฐาน ISO/IEC 27701

4.3.1 การจัดการข้อมูลส่วนบุคคล

นโยบาย

- ข้อมูลส่วนบุคคลที่องค์กรจัดเก็บ ต้องได้รับการปกป้องจากการเข้าถึงโดยไม่ได้รับอนุญาต รวมถึงการเปิดเผย เปลี่ยนแปลง หรือทำลายข้อมูล
- การเก็บรวบรวมข้อมูลส่วนบุคคลต้องดำเนินการอย่างโปร่งใสและได้รับความยินยอมจากเจ้าของข้อมูล

แนวปฏิบัติ

- ใช้การเข้ารหัสข้อมูล (Encryption) และการควบคุมสิทธิ์ (Access Control) เพื่อปกป้องข้อมูลส่วนบุคคล
- ระบุวัตถุประสงค์ในการเก็บข้อมูลอย่างชัดเจน และแจ้งให้เจ้าของข้อมูลทราบก่อนดำเนินการเก็บข้อมูล

4.3.2 การปกป้องความเป็นส่วนตัวระหว่างการพัฒนาระบบ

นโยบาย

- ระบบงานใหม่หรือระบบที่พัฒนาเพิ่มเติมต้องออกแบบและพัฒนาโดยยึดหลัก Privacy by Design และ Privacy by Default
- การทดสอบระบบต้องไม่นำข้อมูลส่วนบุคคลจริงมาใช้ เว้นแต่จำเป็น และต้องมีมาตรการป้องกัน

แนวปฏิบัติ

- กำหนดให้การออกแบบระบบรองรับการปกป้องข้อมูลส่วนบุคคลตั้งแต่เริ่มต้น
- ใช้ข้อมูลที่ทำให้เป็นนิรนาม (Anonymized Data) หรือข้อมูลปลอม (Pseudonymized Data) ในกระบวนการทดสอบระบบ
- ตรวจสอบการพัฒนาระบบอย่างสม่ำเสมอเพื่อให้มั่นใจว่าข้อมูลส่วนบุคคลไม่ได้รับผลกระทบ

4.3.3 การจัดการสิทธิ์ของเจ้าของข้อมูลส่วนบุคคล

นโยบาย

- องค์กรต้องจัดให้มีระบบหรือกระบวนการที่รองรับคำร้องจากเจ้าของข้อมูล เช่น การขอเข้าถึงข้อมูล การแก้ไขข้อมูล หรือการลบข้อมูล
- ต้องดำเนินการกับคำร้องของเจ้าของข้อมูลภายในระยะเวลาที่กำหนดตามข้อกฎหมาย

แนวปฏิบัติ

- สร้างช่องทางให้เจ้าของข้อมูลสามารถยื่นคำร้องได้สะดวก เช่น ผ่านแบบฟอร์มออนไลน์หรือศูนย์บริการลูกค้า
- ตรวจสอบคำร้องทั้งหมดเพื่อให้แน่ใจว่าดำเนินการตามข้อกำหนดของกฎหมาย

4.3.4 การจัดการเหตุการณ์ด้านความปลอดภัยของข้อมูลส่วนบุคคล

นโยบาย

- องค์กรต้องมีแผนรับมือเหตุการณ์ละเมิดข้อมูลส่วนบุคคล (Data Breach) อย่างเป็นระบบ
- ต้องแจ้งเตือนหน่วยงานกำกับดูแลและเจ้าของข้อมูลทันทีเมื่อเกิดเหตุการณ์ที่ส่งผลกระทบต่อข้อมูลส่วนบุคคล

แนวปฏิบัติ

- จัดตั้งทีมรับมือเหตุการณ์ความปลอดภัย (Incident Response Team) พร้อมขั้นตอนปฏิบัติที่ชัดเจน
- ดำเนินการวิเคราะห์สาเหตุของเหตุการณ์ และปรับปรุงกระบวนการป้องกันเพื่อป้องกันเหตุการณ์ซ้ำ

4.3.5 การควบคุมข้อมูลในการจ้างบุคคลภายนอก

นโยบาย

- การจ้างบุคคลภายนอกต้องมีสัญญาที่กำหนดข้อกำหนดด้านความมั่นคงปลอดภัยของข้อมูลส่วนบุคคลอย่างชัดเจน
- ต้องตรวจสอบการปฏิบัติงานของผู้รับจ้างอย่างต่อเนื่อง

แนวปฏิบัติ

- ระบุข้อกำหนดเกี่ยวกับการปกป้องข้อมูลส่วนบุคคลในสัญญากับผู้รับจ้าง
- ดำเนินการตรวจสอบหรือการประเมินความปลอดภัยของบุคคลภายนอกที่จัดการข้อมูลส่วนบุคคล

4.3.6 การทบทวนและปรับปรุงนโยบาย

นโยบาย

- นโยบายและแนวปฏิบัติต้องได้รับการทบทวนอย่างสม่ำเสมอ เพื่อให้สอดคล้องกับการเปลี่ยนแปลงของกฎหมายหรือข้อกำหนดด้านความเป็นส่วนตัว

แนวปฏิบัติ

- จัดให้มีการประชุมเพื่อทบทวนกระบวนการจัดการข้อมูลส่วนบุคคลอย่างน้อยปีละ 1 ครั้ง
- ปรับปรุงนโยบายเมื่อมีการเปลี่ยนแปลงด้านกฎหมาย เช่น การบังคับใช้กฎหมายลูกของ PDPA

5. การทบทวน/แก้ไข

ทบทวนเพื่อการปรับปรุงปีละ 1 ครั้ง หรือมีการเปลี่ยนแปลงที่เกี่ยวกับด้านสารสนเทศในเรื่องการพัฒนาระบบสารสนเทศ

6. เอกสารอ้างอิง

NIST Secure Software Development Framework (SSDF)

ISO/IEC 27001:2022

ISO/IEC 27701:2019