เอกสารการแจ้งเตือนตรวจสอบเว็บไซต์เพื่อป้องกันภัยคุกคามทางไซเบอร์

ศูนย์ประสานการรักษาความมั่นคงปลอดภัยระบบคอมพิวเตอร์แห่งชาติ (ThaiCERT) ขอให้หน่วยงานดำเนินการตรวจสอบเว็บไซต์และปรับปรุงให้มีความปลอดภัยมากยิ่งขึ้น โดยสามารถปรับใช้คำแนะนำตาม OWASP Top Ten 2021[1] โดยดำเนินการตามแนวทางการแก้ไข 10 รายการ ดังนี้

1. การควบคุมสิทธิ์ที่ไม่เหมาะสม (Broken Access Control)

ตัวอย่างปัญหา: ผู้ใช้งานทั่วไปสามารถเข้าถึงหน้า Admin ได้โดยไม่มีสิทธิ์ที่เหมาะสม

วิธีแก้ไข

- ตรวจสอบและจัดการการเข้าถึง (Access Control) ในทุกส่วนของแอปพลิเคชัน
- ใช้การยืนยันตัวตน (Authentication) และการควบคุมสิทธิ์ (Authorization) อย่างถูกต้อง เช่น แยกระดับสิทธิ์ผู้ใช้ทั่วไปและผู้ดูแลระบบ
- ใช้หลักการ "least privilege" โดยให้สิทธิ์เฉพาะที่จำเป็นสำหรับการทำงานของผู้ใช้

2. การเข้ารหัสและปกป้องข้อมูลที่ไม่เพียงพอ (Cryptographic Failures)

ตัวอย่างปัญหา: รหัสผ่านถูกเก็บเป็นข้อความธรรมดาในฐานข้อมูล

วิธีแก้ไข

- เข้ารหัสข้อมูลที่ละเอียดอ่อน เช่น รหัสผ่าน ด้วยเทคนิคการเข้ารหัสที่แข็งแกร่ง เช่น การเข้ารหัสข้อมูลแบบ Bcrypt หรือ Argon2 เพื่อเพิ่มความปลอดภัย
- ใช้ TLS/SSL เพื่อเข้ารหัสข้อมูลที่ส่งผ่านเครือข่าย ป้องกันการดักจับข้อมูลระหว่างทาง
- หลีกเลี่ยงการใช้การเข้ารหัสที่ล้าสมัย เช่น MD5 หรือ SHA-1 ที่มีช่องโหว่ด้านความปลอดภัย

3. การแทรกคำสั่ง (Injection) SQL Injection, Cross-site scripting (XSS) หรือ Command Injection

ตัวอย่างปัญหา: ผู้ใช้สามารถใส่คำสั่ง SQL ลงในช่องค้นหาแล้วดึงข้อมูลทั้งหมดออกมา

วิธีแก้ไข

- ใช้ Prepared Statements หรือ Parameterized Queries สำหรับการเชื่อมต่อฐานข้อมูลเพื่อป้องกันการใส่คำสั่ง SQL ที่ไม่พึงประสงค์
- ตรวจสอบและกรองข้อมูลที่ได้รับจากผู้ใช้ (Input Validation) เพื่อป้องกันการใส่โค้ดที่อันตราย
- จำกัดสิทธิ์ของบัญชีที่เชื่อมต่อกับฐานข้อมูล (Database Account Privilege) ให้สามารถทำงานเฉพาะที่จำเป็นเท่านั้น

4. การออกแบบที่ไม่ปลอดภัย (Insecure Design)

ตัวอย่างปัญหา: แอปพลิเคชันไม่ได้ออกแบบเพื่อป้องกันการโจมตีแบบ Cross-site Scripting (XSS)

วิธีแก้ไข

- ใช้การเข้ารหัสข้อมูล (Data Encoding) ก่อนนำข้อมูลจากผู้ใช้มาแสดงผลในหน้า HTML เพื่อป้องกันการโจมตีแบบ XSS
- ออกแบบระบบตามแนวคิด Secure by Design โดยเพิ่มการตรวจสอบและการป้องกันความปลอดภัยทุกขั้นตอนการพัฒนา
- ใช้ Threat Modeling เพื่อตรวจสอบการออกแบบระบบล่วงหน้า คาดการณ์และป้องกันรูปแบบการโจมตีที่อาจจะเกิดขึ้น

5. การจัดการการกำหนดค่าที่ผิดพลาด (Security Misconfiguration)

ตัวอย่างปัญหา: เซิร์ฟเวอร์เว็บยังคงใช้ค่าตั้งต้นของผู้ดูแลระบบ เช่น ชื่อผู้ใช้และรหัสผ่านเริ่มต้น

วิธีแก้ไข

- ปิดการใช้งานฟีเจอร์ที่ไม่จำเป็น เช่น การแสดงรายละเอียดข้อผิดพลาด (Error Messages) ในระบบ Production เพื่อป้องกันข้อมูลสำคัญรั่วไหล

- เปลี่ยนการตั้งค่าดั้งเดิม (Default Configuration) เช่น รหัสผ่านเริ่มต้น หรือสิทธิ์การเข้าถึง เพื่อป้องกันผู้ไม่หวังดีเข้าถึงระบบ

- ทำการตรวจสอบความปลอดภัยอย่างสม่ำเสมอ เพื่อให้แน่ใจว่าการตั้งค่าเป็นไปตามมาตรฐานความปลอดภัย

6. ซอฟต์แวร์ล้าสมัยและไม่ได้รับการอัปเดต (Vulnerable and Outdated Components)

ตัวอย่างปัญหา: ใช้ไลบรารีที่มีช่องโหว่ เช่น การใช้เวอร์ชันเก่าของ jQuery ที่มีประกาศด้านความปลอดภัย

วิธีแก้ไข

- อัปเดตซอฟต์แวร์และไลบรารีของบุคคลที่สามให้เป็นเวอร์ชันล่าสุดที่ปลอดภัยอย่างสม่ำเสมอ

- ตรวจสอบรายการช่องโหว่ที่เกี่ยวข้องกับส่วนประกอบที่ใช้งาน เช่น การรายงานจาก CVE (Common Vulnerabilities and Exposures) เพื่อทราบถึงความเสี่ยงและดำเนินการแก้ไขทันที

7. การยืนยันตัวตนที่ไม่ปลอดภัย (Identification and Authentication Failures)

ตัวอย่างปัญหา: ผู้โจมตีสามารถแอบอ้างตัวตนของผู้ใช้งานได้โดยการขโมยข้อมูลเซสชัน

วิธีแก้ไข

- ใช้การเข้ารหัสข้อมูลของเซสชัน (Session Encryption) เพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาตและกำหนดให้ใช้คุกกี้ที่ปลอดภัย (Secure Cookies) เพื่อลดความเสี่ยงจากการขโมยข้อมูลเซสชัน

- นำระบบการยืนยันตัวตนแบบหลายปัจจัย (Multi-Factor Authentication MFA) มาใช้เพื่อเพิ่มระดับความปลอดภัยในการยืนยันตัวตนและป้องการแอบอ้าง

- กำหนดเวลาหมดอายุของเซสชัน (Session Timeout) เพื่อให้เซสชันหมดอายุโดยอัตโนมัติเมื่อไม่มีการใช้งานเป็นเวลานาน

- ใช้เทคนิคในการทำให้เซสชันไม่สามารถคาดเดา หรือขโมยได้ง่าย เช่น การสร้างรหัสเซสชันที่ซับซ้อนและเปลี่ยนแปลงรหัสเซสชันเป็นระยะ ๆ เพื่อเพิ่มความปลอดภัย

8. ความปลอดภัยของซอฟต์แวร์และข้อมูลที่ไม่ได้รับการตรวจสอบ (Software and Data Integrity Failures)

ตัวอย่างปัญหา: ไม่มีการตรวจสอบความสมบูรณ์ของไฟล์ที่ดาวน์โหลดมาจากภายนอก

วิธีแก้ไข

- ใช้การลงลายมือชื่อดิจิทัล (Digital Signature) หรือการตรวจสอบค่า Hash เพื่อยืนยันความสมบูรณ์และความถูกต้องของไฟล์ที่ดาวน์โหลด หรือได้นำเข้า

- ตรวจสอบและประเมินความถูกต้องของข้อมูลที่ได้รับจากแหล่งที่ไม่น่าเชื่อถือ เพื่อลดความเสี่ยงจากการใช้ข้อมูลที่อาจถูกดัดแปลง หรือมีช่องโหว่ด้านความปลอดภัย

9. การบันทึกและการตรวจสอบความปลอดภัยที่ไม่เพียงพอ (Security Logging and Monitoring Failures)

ตัวอย่างปัญหา: แอปพลิเคชันไม่สามารถบันทึกเหตุการณ์ที่สำคัญ เช่น ความพยายามในการล็อกอินที่ไม่สำเร็จ

วิธีแก้ไข

- เปิดใช้งานการบันทึกเหตุการณ์สำคัญ (Logging) และทำการตรวจสอบข้อมูลอย่างสม่ำเสมอ เพื่อให้สามารถติดตามและวิเคราะห์เหตุการณ์ที่เกิดขึ้นได้
- ใช้เครื่องมือการตรวจจับและตอบสนองต่อการโจมตี เช่น SIEM (Security Information and Event Management) เพื่อเสริมสร้างความสามารถในการตรวจสอบและจัดการเหตุการณ์ด้านความปลอดภัย
- กำหนดการแจ้งเตือนเมื่อเกิดเหตุการณ์ที่ผิดปกติ เพื่อให้สามารถตอบสนองต่อเหตุการณ์ที่อาจส่งผลกระทบต่อความปลอดภัยได้อย่างทันท่วงที

10. การปลอมแปลงคำขอฝั่งเซิร์ฟเวอร์ไปยังเครือข่ายที่มีการจำกัดการเข้าถึง (Server-Side Request Forgery - SSRF)

ตัวอย่างปัญหา: แอปพลิเคชันสามารถดึงข้อมูลจาก URL ภายในเครือข่ายได้ตามคำขอจากผู้ใช้ภายนอก

วิธีแก้ไข

- จำกัดการเข้าถึงทรัพยากรภายใน (Internal Resources) ผ่านเครือข่าย เพื่อป้องกันการเข้าถึงข้อมูลที่ไม่พึงประสงค์
- ตรวจสอบและกรอง URL ที่ผู้ใช้สามารถร้องขอผ่านแอปพลิเคชัน เพื่อให้แน่ใจว่าไม่มีการเข้าถึง URL ที่ไม่ปลอดภัย
- ใช้ Allowlist สำหรับ URL ที่อนุญาตให้แอปพลิเคชันสามารถดึงข้อมูลได้ โดยการกำหนดเฉพาะ URL ที่เชื่อถือได้และจำเป็นเท่านั้น

ทั้งนี้ สามารถติดตามข่าวสารเกี่ยวกับภัยคุกคามทางไซเบอร์เพิ่มเติมได้ที่ https://webboard-nsoc.ncsa.or.th/ หรือ Scan QR Code

https://webboard-nsoc.ncsa.or.th/

อ้างอิง

1. https://owasp.org/Top10/