โรงพยาบาลดอนมดแดง

DONMODDAENG HOSPITAL

รายงานการทดสอบเจาะระบบ

(Penetration Test Executive Report)

ของ

โรงพยาบาลดอนมดแดง

วันที่ออกรายงาน 15 พฤษภาคม พ.ศ.2569 ที่โรงพยาบาลดอนมดแดง

สารบัญ

1. ข้อมูลทั่วไป (General Information)

ชื่อหน่วยงาน : โรงพยาบาลดอนมดแดง จังหวัดอุบลราชธานี

ระยะเวลาการทดสอบ: 12 พฤษภาคม 2569 – 15 พฤษภาคม 2569

ทีมผู้ทดสอบ: กลุ่มการประกันยุทธศาสตร์ฯ โรงพยาบาลดอนมดแดง

วันที่ออกรายงาน: 15 พฤษภาคม 2569

ประเภทเอกสาร: ลับมาก

2. บทสรุปสำหรับผู้บริหาร (Executive Summary)

2.1 วัตถุประสงค์การทดสอบ

การตรวจสอบช่องโหว่ (Vulnerability Assessment Scan) และการทดสอบเจาะระบบ (Penetration Testing) เพื่อให้ทราบถึงความเสี่ยง จุดอ่อน และระดับความรุนแรง พร้อมรับมือและดำเนินการจัดการความเสี่ยงที่พบได้อย่างเหมาะสม และยกระดับความมั่นคงปลอดภัยทางไซเบอร์ให้สอดคล้องกับเกณฑ์การประเมินการรักษาความมั่นคงปลอดภัยไซเบอร์ (CTAM+: Cybersecurity Technical Assessment Matrix Plus)

2.2 ขอบเขตการทดสอบ

ระบบเป้าหมาย: เว็บไซต์ ที่ URL : https://www.dmdhospital.go.th

ช่วง IP Address: 103.246.18.14 Port: [80,443]

ข้อจำกัด: มีข้อจำกัดด้านเวลา การทดสอบเจาะระบบเป็นการประเมินความปลอดภัยของระบบ ณ ช่วงเวลาที่ทำการทดสอบเท่านั้น

2.3 เครื่องมือที่ใช้ในการทดสอบ

เครื่องมือ (tool)	เวอร์ชัน (version)
Zed Attack Proxy (ZAP)	2.17.0
Kali Linux	2026.1

2.4 การประเมินโดยรวม

CVSS (Common Vulnerability Scoring System) โดยอิงจากคะแนน CVSS v3.x ซึ่งใช้คะแนนตั้งแต่ 0.0 ถึง 10.0 เพื่อจัดระดับความรุนแรงของช่องโหว่

วิกฤต (Critical) / สูง (High) / ปานกลาง (Medium) / ความเสี่ยงต่ำ (Low)

ระดับความรุนแรง (Severity Level)	ช่วงคะแนน (CVSS Score Range)	คำอธิบาย (Description)
Critical (วิกฤต)	9.0 – 10.0	การใช้ช่องโหว่นี้ จะทำให้ผู้โจมตีได้รับสิทธิ์เข้าถึงระดับผู้ดูแลระบบ ไปยังระบบและหรือเข้าถึงข้อมูลที่มีความสำคัญสูง ซึ่งจะส่งผลกระทบร้ายแรงต่อองค์กร ช่องโหว่ที่ระบุว่าเป็น CRITICAL (วิกฤติ) ควรพิจารณาดำเนินการแก้ไขทันที
High (สูง)	7.0 – 8.9	การใช้ช่องโหว่นี้ ทำให้สามารถเข้าถึงข้อมูลที่มีมูลค่าสูงได้ อย่างไรก็ตาม มีข้อกำหนดเบื้องต้นบางอย่างที่ต้องเป็นไปตามเงื่อนไข เพื่อให้การโจมตีประสบความสำเร็จช่องโหว่เหล่านี้ ควรได้รับการพิจารณาแก้ไขให้เร็วที่สุด
Medium (ปานกลาง)	4.0 – 6.9	การใช้ช่องโหว่นี้ อาจขึ้นอยู่กับปัจจัยภายนอกหรือเงื่อนไขอื่นๆ ที่ทำได้ยาก เช่น การต้องได้รับสิทธิ์ผู้ใช้ก่อนจึงจะโจมตีได้สำเร็จ สิ่งเหล่านี้เป็นปัญหาด้านความปลอดภัยระดับปานกลาง ควรได้รับการพิจารณาแก้ไขให้เร็วที่สุดเท่าที่จะทำได้
Low (ต่ำ)	0.1 – 3.9	ช่องโหว่ในระดับต่ำมักมีผลกระทบต่อการดำเนินธุรกิจขององค์กรน้อยมาก การใช้ช่องโหว่ดังกล่าวโดยทั่วไปมักจะต้องเข้าถึงระบบในเครื่อง (local) หรือเข้าถึงทางกายภาพ (physical system access) ควรได้รับการพิจารณาแก้ไข

OWASP Top 10 (2021)

OWASP Top 10 (2021)	ความรุนแรง	จำนวน	ความรุนแรง	จำนวน
A01 Broken Access Control	Medium	1		
A02 Cryptographic Failures				
A03 Injection				
A04 Insecure Design				
A05 Security Misconfiguration	Medium	2		
A06 Vulnerable and Outdated Components	High (สูง)	1		
A07 Identification and Authentication Failures				
A08 Software and Data Integrity Failures	Medium	1		
A09 Security Logging and Monitoring Failures				
A10 Server-Side Request Forgery (SSRF)				

3. สรุปช่องโหว่ที่พบ (Vulnerability Summary)

Penetration Test Report of Donmoddeang Hospital

Site: https://dmdhospital.go.th

Generated on Tue, 19 May 2026 12:59:26

ZAP Version: 2.17.0

ZAP by Checkmarx

Summary of Alerts

Risk Level	Number of Alerts
High	1
Medium	4

Alerts

Name	Risk Level	Number of Instances
Vulnerable JS Library	High	1
Absence of Anti-CSRF Tokens	Medium	Systemic
Content Security Policy (CSP) Header Not Set	Medium	Systemic
Missing Anti-clickjacking Header	Medium	Systemic
Sub Resource Integrity Attribute Missing	Medium	Systemic

4. ช่องโหว่สำคัญและผลกระทบ (Critical Findings & Business Impact)

4.1 Vulnerable JS Library

ระดับความรุนแรง (Severity): High

คำอธิบาย

ช่องโหว่นี้เกิดจากการใช้ไลบรารี JavaScript ของบุคคลที่สามที่ล้าสมัยและมีรายงานข้อบกพร่องด้านความปลอดภัย จากภาพระบุชัดเจนว่าเป็นไลบรารีชื่อ "Swiper" เวอร์ชัน 8.4.5 ซึ่งถูกเรียกใช้งานผ่านปลั๊กอิน Elementor ในระบบ WordPress (ตาม URL ของไฟล์ swiper.min.js ที่ปรากฏ) เครื่องมือ Retire.js ตรวจพบว่าไฟล์เวอร์ชันนี้ตรงกับฐานข้อมูลช่องโหว่ (CVE/GHSA) ที่ถูกเปิดเผยออกมาแล้ว จึงแจ้งเตือนให้ทราบว่าระบบกำลังพึ่งพาสคริปต์ที่ไม่ปลอดภัย

ภาพที่ 2 แสดงผลการทดสอบช่องโหว่ Vulnerable JS Library

ผลกระทบ

การใช้งานไลบรารี JavaScript ที่มีช่องโหว่มักนำไปสู่ความเสี่ยงฝั่งไคลเอนต์ (Client-side attacks) โดยเฉพาะการโจมตีประเภท Cross-Site Scripting (XSS) หากผู้ไม่หวังดีสามารถใช้ประโยชน์จากบั๊กใน Swiper เวอร์ชันนี้ได้ พวกเขาอาจสามารถรันสคริปต์อันตรายบนเบราว์เซอร์ของผู้ที่เข้ามาเยี่ยมชมเว็บไซต์ ซึ่งอาจนำไปสู่การขโมยข้อมูลเซสชัน (Session Hijacking) การหลอกลวงให้ผู้ใช้กรอกข้อมูลสำคัญ หรือการเปลี่ยนหน้าเว็บไปยังเว็บไซต์อันตราย ทำให้เว็บไซต์สูญเสียความน่าเชื่อถือ

แนวทางการแก้ไข

ระบบได้แนะนำแนวทางที่ตรงจุดที่สุดคือ "Upgrade to the latest version of the affected library" (อัปเกรดเป็นเวอร์ชันล่าสุด) สำหรับกรณีนี้ เนื่องจากไฟล์มาจากปลั๊กอิน Elementor วิธีที่ปลอดภัยและดีที่สุดคือการเข้าไปกด อัปเดตปลั๊กอิน Elementor ในระบบหลังบ้านของ WordPress ให้เป็นเวอร์ชันล่าสุด ซึ่งผู้พัฒนาปลั๊กอินมักจะอัปเดตไลบรารีภายในให้ปลอดภัยแล้ว หลังจากอัปเดตควรทดสอบหน้าเว็บอีกครั้งเพื่อตรวจสอบว่าการแสดงผลยังทำงานได้ปกติและไม่มีอะไรพังจากการเปลี่ยนเวอร์ชัน

การจำแนกความเสี่ยง (Vulnerability classifications)

CWE-1395: (Dependency on Vulnerable Third-Party)

4.2 Absence of Anti-CSRF Tokens

ระดับความรุนแรง (Severity): Medium

คำอธิบาย

ช่องโหว่นี้คือ Absence of Anti-CSRF Tokens หรือการที่แบบฟอร์มรับข้อมูลไม่มีระบบป้องกันการโจมตีแบบ CSRF (Cross-Site Request Forgery) การโจมตีประเภทนี้คือการที่แฮกเกอร์สร้างลิงก์หลอกลวงขึ้นมา เมื่อผู้ใช้ (ที่ล็อกอินเข้าระบบเว็บอยู่แล้ว) เผลอไปคลิก เบราว์เซอร์จะส่งคำสั่งไปยังเซิร์ฟเวอร์โดยอัตโนมัติ ทำให้เกิดการเปลี่ยนแปลงข้อมูลหรือส่งฟอร์มโดยที่ผู้ใช้ไม่ได้ตั้งใจกระทำเอง

Absence of Anti-CSRF Tokens
URL: https://dmdhospital.go.th/?page_id=1184
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: <form id="wpforms-form-1177" class="wpforms-validate wpforms-form wpforms-ajax-form" data-formid="1177" method="post" enctype="multipart/form-data" action="/?page_id=1184" data-token="e6a6b16096e0e6aa3dadb40e9de3e800" data-token-time="1779155468">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 - Absence of Anti-CSRF Tokens)
Input Vector:

Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be.

Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, _csrfToken] was found in the following HTML form: [Form 2: "page_id" "page_title" "page_url" "url_referer" "wpforms-1177-field_1" "wpforms-1177-field_2" "wpforms-1177-field_4" "wpforms-1177-field_5" "wpforms[id]" "wpforms[post_id]"].

Solution:
Phase: Architecture and Design
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Reference:
https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://cwe.mitre.org/data/definitions/352.html

ภาพที่ 3 แสดงผลการทดสอบช่องโหว่ Absence of Anti-CSRF Tokens

แนวทางการแก้ไข (Solution)

หลักการแก้ไขคือต้องแน่ใจว่าระบบมีการสร้างและส่งค่า Token แบบสุ่ม (Anti-CSRF Token) ทุกครั้งที่มีการส่งฟอร์ม เพื่อยืนยันว่าคำขอนั้นมาจากหน้าฟอร์มจริงๆ ไม่ใช่ลิงก์ปลอม สำหรับในภาพที่เป็นการใช้ปลั๊กอิน WPForms บน WordPress แนะนำให้ตรวจสอบว่าปลั๊กอินอัปเดตเป็นเวอร์ชันล่าสุดแล้ว และได้เปิดใช้งานฟีเจอร์ Anti-Spam หรือ Token ป้องกันในหน้าการตั้งค่าของตัวปลั๊กอินอย่างถูกต้องครับ

การจำแนกความเสี่ยง (Vulnerability classifications)

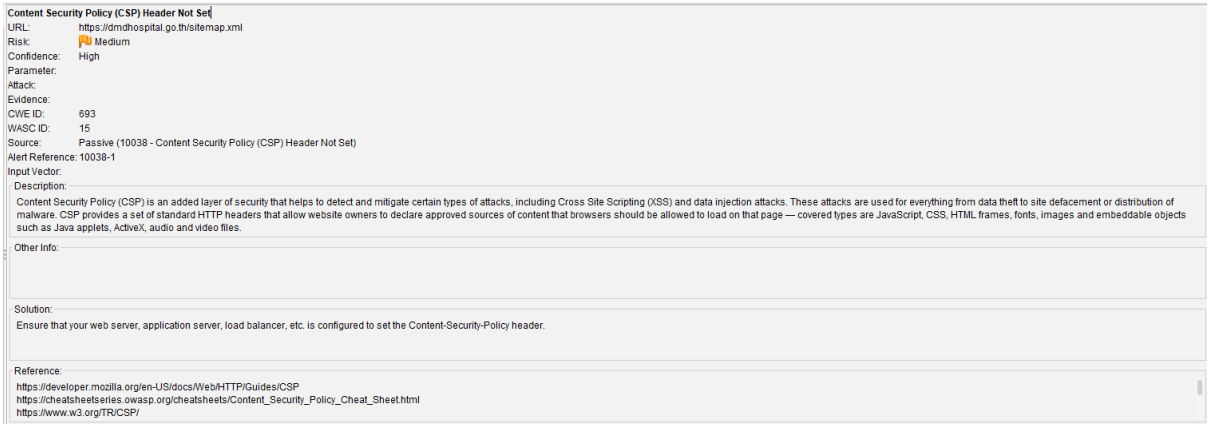
CWE-352: Cross-Site Request Forgery (CSRF)

4.3 Content Security Policy (CSP) Header Not Set

ระดับความรุนแรง (Severity): Medium

คำอธิบาย

ช่องโหว่นี้คือ **Content Security Policy (CSP) Header Not Set** หรือการที่เว็บเซิร์ฟเวอร์ไม่ได้กำหนดนโยบายความปลอดภัยของเนื้อหา (CSP) ไว้ในส่วน HTTP Header ซึ่ง CSP เปรียบเสมือน "บัญชีอนุญาต" (Whitelist) ที่บอกเบราว์เซอร์ว่าเว็บไซต์นี้อนุญาตให้โหลดทรัพยากร (เช่น JavaScript, CSS, รูปภาพ) จากแหล่งใดได้บ้าง เมื่อไม่ได้ตั้งค่าส่วนนี้ไว้ เบราว์เซอร์ก็จะอนุญาตให้รันโค้ดจากทุกแหล่งโดยปริยาย


Content Security Policy (CSP) Header Not Set
URL: https://dmdhospital.go.th/sitemap.xml
Risk: Medium
Confidence: High
Parameter:
Attack:
Evidence:
CWE ID: 693
WASC ID: 15
Source: Passive (10038 - Content Security Policy (CSP) Header Not Set)
Alert Reference: 10038-1
Input Vector:
Description:
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
Other Info:
Solution:
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
Reference:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/

ภาพที่ 4 แสดงผลการทดสอบช่องโหว่ Content Security Policy (CSP) Header Not Set

แนวทางการแก้ไข (Solution)

วิธีการแก้ไขคือต้องตั้งค่าที่ตัวเว็บเซิร์ฟเวอร์ (เช่น Apache, Nginx) หรือกำหนดในฝั่งโค้ดของแอปพลิเคชัน เพื่อให้ส่งค่า Content-Security-Policy กลับไปพร้อมกับการตอบกลับทุกครั้ง โดยควรกำหนดกฎให้รัดกุม เช่น อนุญาตให้โหลดสคริปต์จากโดเมนของโรงพยาบาลดอนมดแดงเองเท่านั้น (default-src 'self') ทั้งนี้ควรทดสอบการใช้งานอย่างละเอียดหลังตั้งค่า เพื่อไม่ให้บล็อกการทำงานของปลั๊กอินหรือฟอนต์ภายนอกที่จำเป็นต้องใช้

การจำแนกความเสี่ยง (Vulnerability classifications)

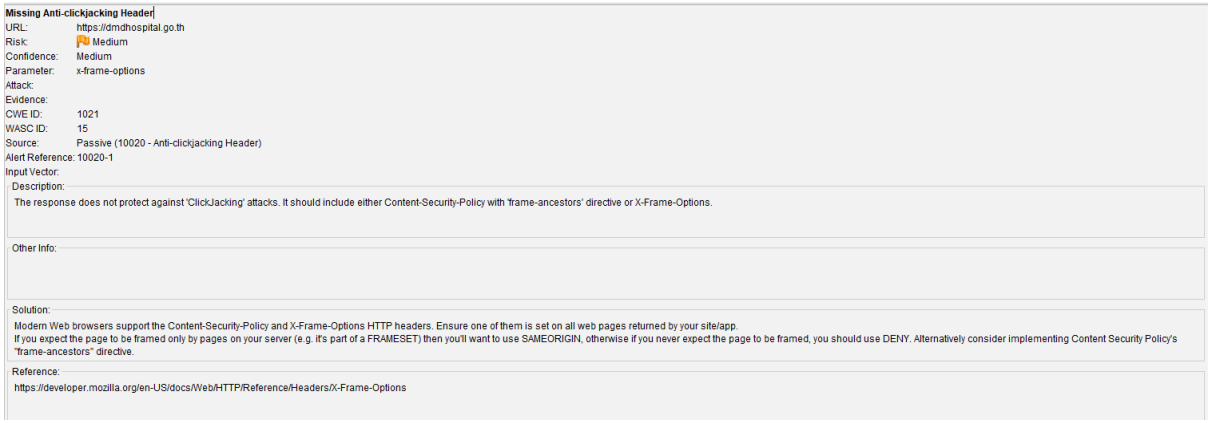
CWE-693: Protection Mechanism Failure

4.4 Missing Anti-clickjacking Header

ระดับความรุนแรง (Severity): Medium

คำอธิบาย

ช่องโหว่นี้คือการขาดการตั้งค่าป้องกันการโจมตีแบบ **Clickjacking** (การลักลอบคลิก) ตามมาตรฐานความปลอดภัย เว็บเซิร์ฟเวอร์ควรส่ง HTTP Header อย่าง X-Frame-Options หรือ Content-Security-Policy (CSP) เพื่อบอกเบราว์เซอร์ว่าเว็บไซต์นี้อนุญาตให้นำไปแสดงผลซ้อนอยู่ในกรอบ (iframe) ของเว็บไซต์อื่นได้หรือไม่ การที่ระบบไม่มีการตั้งค่าส่วนนี้ แสดงว่าหน้าเว็บของโรงพยาบาลสามารถถูกดึงไปครอบหรือแสดงผลบนโดเมนอื่นได้โดยอิสระ


Missing Anti-clickjacking Header
URL: https://dmdhospital.go.th
Risk: Medium
Confidence: Medium
Parameter: x-frame-options
Attack:
Evidence:
CWE ID: 1021
WASC ID: 15
Source: Passive (10020 - Anti-clickjacking Header)
Alert Reference: 10020-1
Input Vector:
Description:
The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.
Other Info:
Solution:
Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
Reference:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-Options

ภาพที่ 5 แสดงผลการทดสอบช่องโหว่ Missing Anti-clickjacking Header

แนวทางการแก้ไข (Solution)

วิธีแก้ไขคือการปรับตั้งค่าเว็บเซิร์ฟเวอร์ให้ส่ง Header ป้องกันกลับมาเสมอ โดยสามารถเลือกใช้ X-Frame-Options แล้วกำหนดค่าเป็น SAMEORIGIN (เพื่อให้แสดงผลใน iframe ได้เฉพาะหน้าเว็บที่มาจากโดเมนเดียวกันเท่านั้น) หรือตั้งค่าเป็น DENY (ห้ามแสดงผลใน iframe เด็ดขาด) นอกจากนี้ แนะนำให้ใช้มาตรฐานใหม่ที่ยืดหยุ่นกว่าคือ Content-Security-Policy พร้อมระบุพารามิเตอร์ frame-ancestors เพื่อจำกัดสิทธิ์ให้รัดกุมยิ่งขึ้นครับ

การจำแนกความเสี่ยง (Vulnerability classifications)

CWE-1021: Improper Restriction of Rendered UI Layers or Frames

4.5 Sub Resource Integrity Attribute Missing

ระดับความรุนแรง (Severity): Medium

คำอธิบาย

ช่องโหว่นี้เกิดขึ้นเมื่อเว็บไซต์มีการดึงไฟล์สคริปต์ (JavaScript) หรือสไตล์ชีต (CSS) จากเซิร์ฟเวอร์ภายนอก (เช่น CDN) มาใช้งาน แต่ไม่ได้แนบค่ารหัสแฮช (Hash) ผ่านแอตทริบิวต์ที่เรียกว่า integrity (Subresource Integrity - SRI) ทำให้เบราว์เซอร์ไม่สามารถตรวจสอบได้ว่าไฟล์ที่ถูกส่งมาจากเซิร์ฟเวอร์ภายนอกนั้นเป็นไฟล์ต้นฉบับที่ถูกต้องจริงๆ หรือถูกผู้ไม่หวังดีแอบดัดแปลงแก้ไขระหว่างทางหรือไม่

ภาพที่ 5 แสดงผลการทดสอบช่องโหว่ Sub Resource Integrity Attribute Missing

แนวทางการแก้ไข (Solution)

วิธีแก้ไขตามมาตรฐานคือการเพิ่มแอตทริบิวต์ integrity (ระบุค่าแฮชของไฟล์ต้นฉบับ) และ crossorigin="anonymous" เข้าไปในแท็ก <script> หรือ <link> ที่เรียกใช้ไฟล์ภายนอก อย่างไรก็ตาม สำหรับบริการที่ไฟล์มีการสร้างใหม่แบบไดนามิกอยู่เสมออย่าง Google Fonts (ตามที่ปรากฏในภาพ) การตั้งค่า SRI อาจทำได้ยากและทำให้ฟอนต์ไม่แสดงผล วิธีที่ปลอดภัยและยั่งยืนกว่าในกรณีนี้คือการดาวน์โหลดไฟล์ฟอนต์มาเก็บและเรียกใช้งานจากเซิร์ฟเวอร์ของเว็บไซต์ (Self-hosted) โดยตรงครับ

การจำแนกความเสี่ยง (Vulnerability classifications)

CWE-345: Insufficient Verification of Data Authenticity

ภาคผนวก

เครื่องมือที่ใช้ในการทดสอบ (Testing Tools)

Zed Attack Proxy (ZAP): เป็นเครื่องมือ ที่ได้รับความนิยมและเป็นที่รู้จักอย่างกว้างขวางในหมู่ผู้เชี่ยวชาญด้านความมั่นคงปลอดภัยทางไซเบอร์ โดยเฉพาะอย่างยิ่งในด้านการ ทดสอบการเจาะระบบเว็บแอปพลิเคชัน (Web Application Penetration Testing) และการประเมินช่องโหว่ของเว็บไซต์ (Web Vulnerability Assessment)

Kali Linux: คือระบบปฏิบัติการ (OS) แบบ Open-source ที่พัฒนาต่อยอดมาจาก Debian โดยถูกออกแบบมาเพื่อการใช้งานด้าน Cybersecurity โดยเฉพาะ ไม่ว่าจะเป็นการทดสอบเจาะระบบ (Penetration Testing), การตรวจสอบความปลอดภัย (Security Auditing) หรือการนิติวิทยาศาสตร์ทางคอมพิวเตอร์ (Computer Forensics)

คำศัพท์และคำอธิบาย

ช่องโหว่ (Vulnerability): คือจุดอ่อนในระบบที่สามารถถูกโจมตีได้

การโจมตี (Attack): ความพยายามในการเข้าถึงหรือทำลายระบบโดยไม่ได้รับอนุญาต

Man-in-the-Middle (MITM): เป็นการโจมตีที่ผู้ไม่หวังดีแทรกตัวเองเข้าไปอยู่ระหว่างการสื่อสารของสองฝ่าย เพื่อดักฟัง แก้ไข หรือขโมยข้อมูลที่ส่งผ่าน

Multi-Factor Authentication: ระบบการยืนยันตัวตนที่ใช้หลายปัจจัย

Black Box Testing: การทดสอบโดยที่ผู้ทดสอบไม่มีข้อมูลเกี่ยวกับระบบเลย

Grey Box Testing: การทดสอบโดยที่ผู้ทดสอบได้รับข้อมูลบางส่วนเกี่ยวกับระบบ

White Box Testing: การทดสอบโดยที่ผู้ทดสอบได้รับข้อมูลทั้งหมดเกี่ยวกับระบ

Common Vulnerabilities and Exposures (CVE): ช่องโหว่และความเสี่ยงที่พบบ่อย โดยพื้นฐานแล้ว CVE คือ รายการสาธารณะที่รวบรวมและจัดหมวดหมู่ช่องโหว่ด้านความมั่นคงปลอดภัยของคอมพิวเตอร์ที่ถูกเปิดเผยต่อสาธารณะ ไม่ว่าจะเป็นช่องโหว่ในซอฟต์แวร์ ฮาร์ดแวร์ เฟิร์มแวร์ หรือระบบปฏิบัติการ

Common Weakness Enumeration (CWE): รายการมาตรฐานสากลที่รวบรวมและจำแนกประเภทของ "จุดอ่อน" (Weaknesses) ในซอฟต์แวร์และฮาร์ดแวร์ โดยมีเป้าหมายเพื่อเป็นภาษาเดียวกันให้นักพัฒนาและผู้เชี่ยวชาญด้านความปลอดภัยใช้สื่อสารกัน

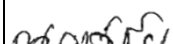
Web Application Firewall (WAF): เป็นระบบป้องกันความปลอดภัยที่ออกแบบมาเฉพาะเพื่อปกป้องเว็บแอปพลิเคชัน

ติดต่อสอบถามข้อมูล

กลุ่มงานประกันสุขภาพและสารสนเทศทางการแพทย์ โรงพยาบาลดอนมดแดง

1 หมู่12 ต.เหล่าแดง อ.ดอนมดแดง จังหวัดอุบลราชธานี 34000

โทร: 045308054 ต่อ 128

หน้าที่	ชื่อ	ตำแหน่ง	ลายมือชื่อ
จัดทำเอกสาร	นายณรงค์ชัย สาระรัตน์	นักวิชาการคอมพิวเตอร์ชำนาญการ	
ผู้ทดสอบระบบ	นายณรงค์ชัย สาระรัตน์	นักวิชาการคอมพิวเตอร์ชำนาญการ	
ผู้ตรวจสอบและผู้อนุมัติ	นายวสุวัตติ์ พบลาภ	ผู้อำนวยการโรงพยาบาล