# ความมั่นคงปลอดภัยระบบเว็บและซอฟต์แวร์
# Software and Web Security

# สารบัญ

# บทที่ 1

## จุดประสงค์ของบทเรียน

**1.1 อธิบายเกี่ยวกับระบบเว็บ**

**1.2 เรียนรู้เกี่ยวกับองค์ประกอบ การทำงานของระบบเว็บ**

**1.3 อธิบายการโจมตีเครื่องแม่ข่ายระบบเว็บ และภัยคุกคามระบบเว็บ**

**1.4 อภิปรายมาตรการความปลอดภัยสำหรับระบบเว็บ**

**1.5 อธิบายวิธีการป้องกันการโจมตีเครื่องแม่ข่ายระบบเว็บ**

**1.6 อธิบายแนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัย**

# บทที่ 2

## คำอธิบายเกี่ยวกับบทเรียน

ความรู้ความเข้าใจเกี่ยวกับระบบเว็บ องค์ประกอบต่าง ๆ และการทำงานของระบบเว็บ ภัยคุกคามและการโจมตีระบบเว็บ วิธีการป้องกันการโจมตีและมาตรการความปลอดภัยสำหรับระบบเว็บ แนวทางและขั้นตอนการพัฒนาระบบ แนวทางและขั้นตอนการพัฒนาระบบสมัยใหม่ แนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัย

# บทที่ 3

## กลุ่มสาระบทเรียน

**3.1 เกี่ยวกับระบบเว็บ องค์ประกอบ และการทำงานของระบบเว็บ**

**3.2 การโจมตีเครื่องแม่ข่ายระบบเว็บ**

**3.3 ภัยคุกคามระบบเว็บและมาตรการความปลอดภัยสำหรับระบบเว็บ**

**3.4 การป้องกันการโจมตีเครื่องแม่ข่ายระบบเว็บ**

**3.5 แนวทางและขั้นตอนการพัฒนาระบบ**

**3.6 การพัฒนาระบบให้มีความมั่นคงปลอดภัย**

# บทที่ 4

# นิยามภายในบทเรียน

## 4.1 Web Application

**Web Application :** เว็บแอปพลิเคชัน หมายความว่า แอปพลิเคชันที่ทำงานบน **Web Server** จากระยะไกลและส่งผลลัพธ์ผ่านเครือข่ายอินเทอร์เน็ต โดยใช้เทคโนโลยี **Web 2.0** สำหรับการสื่อสารระหว่างเครื่องผู้ใช้ ผู้ใช้งาน และผู้ใช้อื่นที่เป็นบุคคลที่สาม

## 4.2 Web Server

**Web Server :** เว็บเซิร์ฟเวอร์ หมายความว่า ซอฟต์แวร์ และฮาร์ดแวร์ที่มีไว้เพื่อส่งมอบเนื้อหาเว็บที่สามารถเข้าถึงได้ผ่านอินเทอร์เน็ต

## 4.3 Website Defacement

**Website Defacement :** การเปลี่ยนแปลงหน้าเว็บไซต์ หมายความว่า วิธีการหรือเทคนิคที่ผู้โจมตีใช้ในการเข้าปรับเปลี่ยนข้อมูลเผยแพร่ต่าง ๆ บนหน้าเว็บไซต์ ผ่านช่องโหว่ต่าง ๆ ของระบบผ่านเครือข่าย โดยมากเป็นการโจมตีที่หวังผลด้านการสูญเสียความน่าเชื่อถือของผู้ให้บริการ

## 4.4 OWASP

**OWASP: Open Web Application Security Project :** มาตรฐานความมั่นคงปลอดภัยที่เป็นระบบเปิดของ Web Application ซึ่งจัดทำขึ้นโดยองค์กรไม่แสวงหาผลกำไร เพื่อส่งเสริมความรู้และแนวทางดำเนินการ Web Application Security เพื่อทำให้ระบบมีความมั่นคงปลอดภัย

## 4.5 Secure Socket Layer(SSL)

**Secure Socket Layer (SSL) :** คือเทคนิคการเข้ารหัสช่องทางการสื่อสาร โดยใช้วิธีการเข้ารหัสแบบ สมมาตร และ อสมมาตร เพื่อทำให้การสื่อสารมีความปลอดภัย ข้อมูลไม่รั่วไหลระหว่างรับส่งผ่านเครือข่าย

# บทที่ 5

## เนื้อหาในบทเรียน

**5.1 เกี่ยวกับระบบเว็บ**

**5.2 องค์ประกอบของเว็บแอปพลิเคชัน**

**5.3 วิธีการทำงานของ Web Applications**

**5.4 ประโยชน์ของเว็บแอปพลิเคชันต่อองค์กร**

**5.5 ผลกระทบของการโจมตีเครื่องแม่ข่ายระบบเว็บ**

**5.6 การโจมตีเครื่องแม่ข่ายระบบเว็บ**

**5.7 การเปลี่ยนหน้าเว็บไซต์ (Website Defacement)**

**5.8 การยึดครองเครื่องแม่ข่ายระบบเว็บ**

**5.9 ประเภทภัยคุกคามระบบเว็บ**

**5.10 มาตรการจัดการภัยคุกคามและการป้องกันการโจมตี**

**5.11 แนวทางการพัฒนาระบบให้มีความมั่นคงปลอดภัย**

**5.12 มาตรฐานความมั่นคงปลอดภัย Web Application**

## 5.1 เกี่ยวกับระบบเว็บ

### Web Application

ระบบเว็บแอปพลิเคชัน **(Web Application)** คือ แอปพลิเคชันที่ทำงานบนเซิร์ฟเวอร์และส่งข้อมูลผ่านอินเตอร์เน็ตมายังผู้รับบริการ ปัจจุบันเว็บแอปพลิเคชันให้บริการอยู่บน **Web 2.0 Technologies** และผู้รับบริการสามารถใช้บริการผ่านบราวเซอร์ โมบายแอปพลิเคชันบางประเภท อุปกรณ์ไอโอที **(IoT)** หรืออื่น ๆ ได้

## 5.2 องค์ประกอบของเว็บแอปพลิเคชัน

### Web Application Components

- **Server compute node**

คือ ส่วนที่ประมวลผลลัพธ์โดยผู้ให้บริการ เช่น **nginx, apache, tomcat** เป็นต้น
ตัวอย่าง ได้แก่

  - **Web application** (ส่วนประกอบที่จำเป็น)
  - **Code** (ส่วนประกอบที่จำเป็น)
  - **Access Control** (ส่วนประกอบที่จำเป็น): **login, session, permission** เป็นต้น

- **ส่วนข้อมูลเพื่อการประมวลผลหรือแสดงผล**

ตัวอย่าง ได้แก่ ข้อมูลประเภทอักษร **(Text)** รูปภาพฐานข้อมูล

- **Client compute node**

คือ ส่วนที่ประมวลผลลัพธ์โดยผู้รับบริการเอง ตัวอย่าง ได้แก่ **JavaScript, Applet, Ajax**

## 5.3 วิธีการทำงานของ Web Applications

### Web Application

**Web Application** ที่ให้บริกามี 2 ลักษณะ คือ

- **Web Application** แบบไม่เปลี่ยนแปลง **(Static Web Applications)** เป็น **Web Application** ที่แสดงข้อมูลคงที่ เช่น **html page** ซึ่งจะไม่มีการประมวลผลที่ซับซ้อน
- **Web Application** แบบเปลี่ยนแปลง **(Dynamic Web Application)** เป็น **Web Application** ที่แสดงผลตามที่ผู้ใช้งานต้องกำร เช่น การเลือกชุดข้อมูล **(list data)**

| แบบไม่เปลี่ยนแปลง (Static Web Applications) | แบบเปลี่ยนแปลง (Dynamic Web Application) |
|---|---|
| รับคำร้องจากผู้ใช้บริการ | รับคำร้องจากผู้ใช้บริการ |
| ตรวจสอบความถูกต้องขอคำร้อง | ตรวจสอบความถูกต้องขอคำร้อง |
| - | ประมวลผลคำร้อง |
| - | สร้างข้อมูลเพื่อตอบสนอง |
| ส่งข้อมูลตอบสนองกลับไปยังผู้ร้องขอ | ส่งข้อมูลตอบสนองกลับไปยังผู้ร้องขอ |

- ผู้ใช้งานเรียกใช้ **Web Application** ผ่านทาง **URL, IP Address** โดยส่วนมากจะผ่าน **Port 80, 443** แต่สามารถเป็น **Port** อื่น ๆ ได้ตามที่ **Web Application** กำหนดไว้
- ในกรณีที่คำร้องขอถูกต้อง **Web Application** รับคำร้องขอจากผู้เรียกใช้งานและประมวลผลข้อมูลและตอบกลับด้วยข้อมูลที่ร้องขอ แต่หากคำร้องขอไม่ถูกต้อง **Web Application** จะตอบกลับด้วยสถานะว่าคำร้องขอไม่ถูกต้อง เช่น **404 File not found, 403 Forbidden**

- แอปพลิเคชันปลายทางที่เครื่องของผู้ร้องขอแสดงผล และ **/** หรือนำข้อมูลจาก **Web Application** ประมวลผลฝั่งเครื่องผู้ร้องขอ

## 5.4 ประโยชน์ของเว็บแอปพลิเคชันต่อองค์กร

### Web Application

ปัจจุบันองค์กรโดยส่วนมากดำเนินธุรกิจด้วยระบบสำรสนเทศ จึงทำให้ระบบสำรสนเทศเป็นเครื่องมือสำคัญของการดำเนินธุรกิจ และ **Web Application** เป็นระบบสารสนเทศประเภทหนึ่งที่ผู้ใช้บริการสามารถเข้าถึงได้จากทุกคน ทุกที่ ทุกเวลา ซึ่งจะเห็นได้ว่าธุรกิจสามารถให้บริการได้ตลอดเวลา **(24x7)**

- เพิ่มศักยภาพของธุรกิจขององค์กร
- เพิ่มโอกาสให้การดำเนินธุรกิจได้ตามเป้าหมายที่กำหนด

## 5.5 ผลกระทบของการโจมตีเครื่องแม่ข่ายระบบเว็บ

### Web Application

ดังนั้นหากระบบ **Web Application** ถูกโจมตี ความเสียหายที่องค์กรจะได้รับมีทั้งที่เป็นความเสียหายโดยตรงและความเสียหายโดยอ้อม ดังนี้

### ผลกระทบของการโจมตีเครื่องแม่ข่ายระบบเว็บ

| ความเสียหายโดยตรง | ความเสียหายโดยอ้อม |
|---|---|
| องค์กรสูญเสียรายได้ | โดนปรับตามกฎหมาย กฎระเบียบ |
| องค์กรสูญเสียลูกค้า | องค์กรสูญเสียชื่อเสียง |

## 5.6 การโจมตีเครื่องแม่ข่ายระบบเว็บ

### Web Application

การโจมตี **Web Application** คือ กิจกรรมที่ทำให้ **Web Application** ถูกละเมิดในด้านความมั่นคงปลอดภัย ได้แก่

- การเข้าถึงโดยไม่ได้รับอนุญาต การเข้าถึงข้อมูลโดยไม่ได้อนุญาต
- การทำให้ **Web Application** ประมวลผลไม่ถูกต้อง
- การทำให้ **Web Application** ไม่สามารถให้บริการได้

| ประเภทการโจมตี | ตัวอย่างการโจมตี |
|---|---|
| การโจมตีเพื่อละเมิดด้าน **Confidentiality** | • **SQL Injection**<br>• **Brust force Attack** |
| การโจมตีเพื่อละเมิดด้าน **Integrity** | • **Web Defacement**<br>• **Change configuration** |
| การโจมตีเพื่อละเมิดด้าน **Availability** | • **DoS**<br>• **DDoS**<br>• **Ransomware** |

## 5.7 การเปลี่ยนหน้าเว็บไซต์ (Website Defacement)

### Website Defacement

การเปลี่ยนหน้าเว็บไซต์เป็นการโจมตีเพื่อละเมิดด้าน **Integrity** โดยทำให้หน้าแสดงผลของ **Web Application** เปลี่ยนแปลงไป โดยอาจจะมีการเปลี่ยนแปลงอย่างชัดเจนหรือการเปลี่ยนแปลงอย่างไม่ชัดเจน

- **การเปลี่ยนแปลงแบบชัดเจน** คือ ทำให้ **Web Application** เปลี่ยนอย่างชัดเจน ซึ่งจะทำให้องค์การเสียหายด้านชื่อเสียง
- **การเปลี่ยนแปลงแบบไม่ชัดเจน** คือ ทำให้ **Web Application** มีข้อมูลอื่น ๆ เพิ่มเติม เช่น การฝัง **script,** การเพิ่มเนื้อหาโฆษณา เป็นต้น

## 5.8 การยึดครองเครื่องแม่ข่ายระบบเว็บ

### Web Servers Compromised

**การยึดครอง Web Application** คือ การละเมิดด้าน **Confidentiality** เนื่องจากโดยทั่วไปผู้ไม่ประสงค์ดีไม่มีสิทธิในการจัดการระบบ **Web Application** แต่ผู้ไม่ประสงค์ดี ได้ใช้วิธีการและเทคนิคต่าง ๆ เพื่อให้ได้ข้อมูล หรือ ได้รับสิทธิ์ในระบบ **Web Application**

การยึดครอง **Web Application** จะทำให้องค์กรได้รับความเสียหาย ทั้งด้านชื่อเสียงหรือธุรกิจขององค์กร ทั้งนี้ เหตุให้ผู้ไม่ประสงค์ดีมี แรงจูงใจ ในการโจมตี ดังนี้

- เพื่อชื่อเสียงจากการยึดครอง
- เพื่อเงินจากการขู่กรรโชกองค์กร

## 5.9 ประเภทภัยคุกคามระบบเว็บ

### Web Application Thrats

| ภัยคุกคามที่กระทำต่อระบบ Web Application | |
|---|---|
| **การเข้าถึงระบบเว็บแอปพลิเคชันข้อมูลโดยไม่ได้รับอนุญาต** | |
| **Directory Traversal** | ผู้โจมตีพยายามสุ่มแฟ้มข้อมูลในระบบ **Web Application** หรือพยายามรวมรวมข้อมูลแฟ้มข้อมูลเพื่อเข้าถึงข้อมูลสำคัญเช่นแฟ้มค่าพารามิเตอร์เป็นต้น |
| **SQL Injection** | ผู้โจมตีพยายามส่งข้อมูลผ่านช่องทางสำหรับบันทึกข้อมูลปกติแต่ดำเนินการเพื่อวัตถุประสงค์ให้ผลลัพธ์เกิดข้อผิดพลาด |
| **Broken Access Control** | ผู้โจมตีพยายามละเมิดการเข้าถึงโดยจะหาจุดที่ไม่มีการควบคุมการเข้าถึงและขยายผลเพื่อให้สามารถเข้าถึงส่วนสำคัญของระบบ |

## 5.9 ประเภทภัยคุกคามระบบเว็บ

### Web Application Thrats

| ภัยคุกคามที่กระทำต่อระบบ Web Application | |
|---|---|
| **การทำให้การประมวลผลผิดพลาด** | |
| **Parameter/Form Tampering** | ผู้โจมตีพยายามเปลี่ยนค่าข้อมูลต่างๆ ระหว่างการส่งข้อมูลกลับไปยังเครื่องแม่ข่ายเพื่อให้การประมวลผลผิดพลาด |
| **Improper Error Handling** | ผู้โจมตีพยายามส่งข้อมูลเพื่อให้ระบบประมวลผิดพลาดโดยต้องการรายละเอียดต่างๆของระบบจากค่าตั้งต้นของการตั้งค่าแสดงความผิดพลาด |
| **Log Tampering** | ผู้โจมตีพยายามเข้าถึง log ของระบบเพื่อทำการแก้ไขลบร่องรอยทำให้log ไม่สามารถนำไปวิเคราะห์เหตุการณ์หรือตรวจสอบด้านนิติวิทยาศาสตร์ไม่ได้ |

## 5.9 ประเภทภัยคุกคามระบบเว็บ

### Web Application Thrats

| ภัยคุกคามที่กระทำต่อระบบ Web Application | |
|---|---|
| **การทำให้ระบบหยุดชะงัก** | |
| **Denial of Service** | ผู้โจมตีพยายามทำให้ส่งคำร้องเพื่อให้ระบบประมวลผลและทำให้ระบบทำงานหนักและหยุดชะงัก |
| **Buffer Overflow** | ผู้โจมตีพยายามส่งข้อมูลที่มีขนาดเกินกว่าขนาดของหน่วยความจำซึ่งจะทำให้ระบบปฏิบัติการล้มเหลวและหยุดชะงัก |
| | |

## 5.9 ประเภทภัยคุกคามระบบเว็บ

### Web Application Thrats

| ภัยคุกคามต่อผู้ใช้บริการ | |
|---|---|
| **การปลอมแปลง หลอกล่อผู้ใช้บริการ** | |
| **Cross-Site Scripting (XSS)** | ผู้โจมตีพยายามฝัง **script** ไว้ใน **URL** ของระบบที่ผู้ใช้งานใช้งานเป็นประจำโดย **script** จะทำหน้าที่ขโมยข้อมูลของผู้ใช้งาน เช่นชื่อผู้ใช้งานรหัสผ่าน, แก้ไขไฟล์ในบราวเซอร์ของผู้ใช้งาน, พาผู้ใช้งานไปยังเว็บไซต์ที่อันตราย |
| **Cross-site Request Forger (CSRF)** | ผู้โจมตีพยายามปลอมแปลงคำสั่งโดยทำให้ผู้ให้บริการเข้าใจว่าเป็นการกระทำของผู้ใช้บริการเองและตอบสนองตามที่ผู้โจมตีต้องการ |
| **Cookie Poisoning** | ผู้โจมตีพยายามเปลี่ยนตัวตนของผู้ใช้งานด้วยวิธีการเปลี่ยนค่าพารามิเตอร์ใน **Cookie** ซึ่งเป็นที่เก็บข้อมูลส่วนตัวสำหรับการเข้าใช้งาน **Web Application** |
| **Cookie Snooping** | ผู้โจมตีพยายามเขาถึง **Cookie** ของผู้ใช้งานและเปลี่ยนข้อมูลต่างๆใน **Cookie** เพื่อให้ผู้ใช้งานกลายเป็นคนอื่นตามที่ผู้โจมตีต้องการเมื่อเข้าใช้งาน **Web Application** |

## 5.10 มาตรการจัดการภัยคุกคามและการป้องกันการโจมตีระบบเว็บ

## Web Application Controls

ภัยคุกคามเป็นกิจกรรมจากภายนอกที่มีเป้าหมายให้ระบบ **Web Application** ซึ่งเป็นปัจจัยที่ระบบ **Web Application** ไม่สามารถท าลดลงได้ เช่น ลดการโจมตีด้วย **SQL Injection,** ลดการโจมตี **DoS** แต่สิ่งที่ระบบ **Web Application** สามารถจัดการได้ คือ การลดช่องโหว่ที่การโจมตีนั้น ๆ จะทำให้ระบบ **Web Application** เสียหายได้ เช่น

- การลดความผิดพลาดจากการประมวลผล โดยตรวจสอบโค้ดของระบบ **Web Application**
- การตรวจสอบข้อมูลนำเข้าก่อนประมวล เพื่อลดความเสียหายจากการโจมตีด้วย **SQL Injection**
- การติดตั้ง **Firewall**, **IPS** เพื่อลดความเสียหายจากการโจมตี **DoS**

ดังนั้น การจัดการความเสียหายจากภัยคุกคามระบบ **Web Application** จะต้องพิจารณาถึงช่องโหว่ สาเหตุของช่องโหว่ เพื่อทำให้ช่องโหว่ลดลง

## 5.11 แนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัย

## Secure Development

การสร้างระบบ **Web Application** จะปลอดภัยเมื่อมีการพิจารณาความต้องการด้านความมั่นคงปลอดภัยสารสนเทศตั้งแต่เริ่มต้นของการดำเนินการ ดังนั้นถ้าต้องการให้ระบบ **Web Application** มีความปลอดภัยจะต้องมีการเพิ่มเติมกิจกรรมด้านความปลอดภัยสารสนเทศในทุกช่วงของการพัฒนา

- **Requirements**
- **Design**
- **Implement**
- **Testing**
- **Maintenance**
- **Disposal**

## 5.11 แนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัย

## Secure Development

| ช่วงของวงจร | การดำเนินการ | การดำเนินการ |
|---|---|---|
| **Requirements** | องค์กรต้องกำหนดความต้องการด้านความปลอดภัยโดยต้องสอดคล้องกับเป้าหมายของบริการ,เป้าหมายทางธุรกิจ,ต้นทุนและผลตอบแทน | ทำให้เรารู้ถึงความต้องการความเป็นไปได้ว่าสามารถทำได้หรือไม่ได้,สามารถตอบความต้องการทางธุรกิจได้หรือไม่ |
| **Design** | องค์กรต้องกำหนดรูปแบบของระบบ **WebApplication** เช่น การแบ่งการเชื่อมต่อระหว่างภายในภายนอก,การแบ่งระดับตามความสำคัญของข้อมูลในระบบเป็นต้น | ทำให้ทราบว่าต้องวางแผนอย่างไรอะไรคือสิ่งที่เป็นช่องโหว่ที่หลงเหลือหากต้องการให้ธุรกิจดำเนินต่อไป |
| **Implement** | องค์กรควรนำมาตรฐานต่างๆมาประยุกต์ใช้ในการพัฒนาระบบเช่น **Secure Coding** สอดคล้องตาม **OWASP** เป็นต้น | ทำให้ผลลัพธ์ของการพัฒนาถูกต้องครบถ้วนปลอดภัยและข้อบกพร่องด้านความปลอดภัยถูกจัดการตั้งแต่เริ่มทำ |

## 5.11 แนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัย

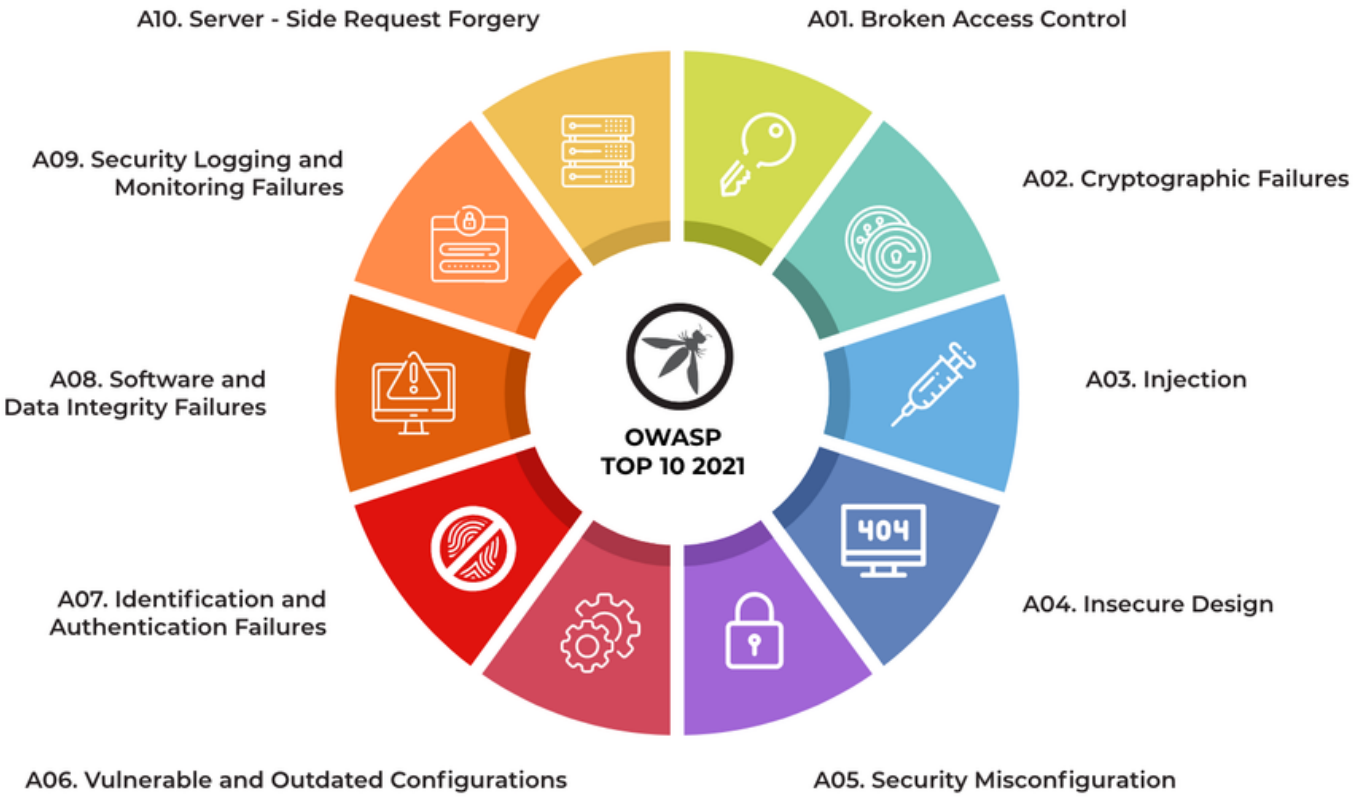
## Secure Development

| ช่วงของวงจร | การดำเนินการ | การดำเนินการ |
|---|---|---|
| **Testing** | องค์กรควรผนวกการตรวจสอบด้าน **non-functional** เช่น **code** มีความปลอดภัยหรือไม่, ตรวจสอบช่องโหว่ก่อนให้บริการ | ทำให้ตรวจพบถึงจุดบกพร่องด้านความปลอดภัยที่คงเหลือทำให้แก้ไขอย่างต่อเนื่องหรือหาวิธีการป้องกันขณะให้บริการได้ |
| **Maintenance** | องค์กรควรติดตามและตรวจสอบช่องโหว่รวมถึงแก้ไขช่องโหว่ใหม่ๆที่เกิดขึ้นในอนาคต | ลดโอกาสที่ระบบจะถูกโจมตีหรือลดความเสียหายหากถูกโจมตี |
| **Disposal** | องค์กรควรมีกระบวนการทำลายหรือลบระบบ **Web application** หรือข้อมูลในนั้นเพื่อให้ไม่สามรถกู้คืนเพื่อทำให้ข้อมูลรั่วไหลได้ | ทำให้ข้อมูลสำคัญต่างๆไม่รั่วไหลแม้จะสิ้นสุดการให้บริการไปแล้ว |

## 5.12 มาตรฐานความมั่นคงปลอดภัยของ Web Application

### Web Security

**OWASP (Open Web Application Security Project)** เป็น มาตรฐานความมั่นคงปลอดภัยของ **Web Application** ซึ่งจัดทำขึ้นโดยองค์กรไม่แสวงหาผลกำไร เพื่อส่งเสริมความรู้และแนวทางดำเนินการ **Web Application Security** เพื่อทำให้ระบบมีความมั่นคงปลอดภัยมากยิ่งขึ้น โดยจะมีผลงานวิจัย รายงาน เอกสาร เครื่องมือ และเทคโนโลยีความมั่นคงปลอดภัยที่แนะนำสำหรับ **Web Application Security** จะมีการปรับปรุงอย่างสม่ำเสมอ ผู้พัฒนาระบบสามารถติดตามและแก้ไขโดยข้อมูลจาก **OWASP** จะทำให้ระบบ **Web Application** มีความปลอดภัยต่อภัยคุกคามตามเทรนได้



## สรุปท้ายบท

- **Web Application** เป็นจุดเชื่อมโยงความสัมพันธ์ระหว่างผู้ใช้งานกับเครื่องแม่ข่ายระบบเว็บ **(Web Servers)** ซึ่งประกอบด้วยชุดคำสั่งต่าง ๆ สำหรับการทำงานของระบบเว็บ

- ภัยคุกคาม **Web Application** เป็นวิธีการ เทคนิค เพื่อละเมิดความปลอดภัยสารสนเทศ **(Confidentiality, Integrity, Availability)** จำแนกเป็น 2 ประเภท ตามเป้าหมายของภัยคุกคาม คือ ระบบ **Web Application** และผู้ใช้บริการ

- การเปลี่ยนหน้าเว็บ **(Web Defacement)** เป็นภัยคุกคามและการโจมตีที่มักเกิดขึ้นบ่อยกับระบบเว็บ

- แนวทางและขั้นตอนการพัฒนาระบบให้มีความมั่นคงปลอดภัยประกอบด้วยกิจกรรมหลัก ได้แก่ **Requirements, Design, Implementation, Testing, Maintenance, Disposal**

- **OWASP (Open Web Application Security Project)** เป็นมาตรฐานความมั่นคงปลอดภัย เพื่อส่งเสริมความรู้และแนวทางดำเนินการ **Web Application Security** ทำให้ระบบมีความมั่นคงปลอดภัย