1. Basic Web Architecture
HTTP คือโปรโตคอลแบบ stateless (แต่ละ request ไม่จำข้อมูลเดิม)
ใช้ URI เพื่อระบุ resource ที่ต้องการเข้าถึง
ประเภทของ HTTP Status Code:
2xx: สำเร็จ
3xx: redirect
4xx: error ฝั่ง client
5xx: error ฝั่ง server
2. Web Cookies และ Sessions
Sessions
- ใช้เพื่อจำสถานะผู้ใช้ (เพราะ HTTP เป็น stateless)
- เก็บ session ได้หลายวิธี:
  - URL (เสี่ยงโดนขโมย)
  - hidden form
  - cookies

Cookies
- ไฟล์เล็กเก็บข้อมูลใน browser ผู้ใช้
- ผู้ใช้สามารถแก้ไขได้ → เสี่ยงถูกเปลี่ยนค่า (cookie poisoning)
- แนะนำให้ใช้ secure, httponly, samesite ป้องกัน

🔷 Cookie + Cryptographic Checksum
- ใช้ key ลับ ตรวจความถูกต้องของ cookie

3. Session Hijacking & Fixation
- การขโมย session ID เพื่อปลอมตัวเป็นผู้ใช้
- สาเหตุหลัก:
  - token คาดเดาง่าย
  - ใช้ PRNG ไม่ปลอดภัย
  - ไม่เปลี่ยน session ID หลัง login
- การป้องกัน:
  - ใช้ HTTPS และเปิด HSTS
  - ใช้ HttpOnly, Secure, SameSite บน cookie
  - regenerate session ID หลัง login

4. OWASP Top 10 (เน้นที่ออกสอบ)
Broken Access Control (BAC)
- ผู้ใช้เข้าถึง resource ที่ไม่ควรได้ เช่น:
  - เข้าถึงข้อมูลคนอื่น (แนว horizontal access)
  - เข้าถึงฟังก์ชัน admin (แนว vertical access)
  - เข้าถึงไฟล์ local เช่น ../../web.config

การป้องกัน:
- ใช้ access control matrix
- ตรวจสอบสิทธิ์ทุก URL และฟังก์ชัน
- จำกัดสิทธิ์ให้เฉพาะผู้มีสิทธิ์เท่านั้น

5. Cross-Site Request Forgery (CSRF)
- หลอกให้ผู้ใช้ที่ login อยู่ส่งคำสั่งไปยังเว็บโดยไม่รู้ตัว เช่นโอนเงิน
- ทำได้ผ่าน <img>, <iframe>, <form>, หรือ JS

การป้องกัน:
1. ตรวจ Referer Header ว่ามาจากเว็บเดียวกัน
2. ใช้ SameSite Cookies (Strict, Lax)
3. ใช้ Secret Token ฝังไว้ในฟอร์มหรือ cookie เปรียบเทียบก่อนดำเนินการ

6. Cross-Site Scripting (XSS)
- โจมตีโดย inject code (ส่วนมากเป็น JavaScript) ลงในเว็บ

ประเภท:
- Reflected XSS: โค้ดวิ่งผ่าน URL และแสดงผลทันที
- Stored XSS: โค้ดถูกเก็บใน server แล้วแสดงผลให้คนอื่นเห็นภายหลัง (เช่น comment)

การป้องกัน:
- กรอง input/output ที่เป็น HTML/JS
- เข้ารหัสพิเศษ เช่น htmlspecialchars() ใน PHP
- ใช้ library เช่น jsoup, OWASP ESAPI

ตัวอย่างการหลบหลีกฟิลเตอร์:
- script ซ่อนใน <iframe>, ใช้ %3c แทน <

📌 3. การโจมตีในระดับ TCP/IP

ประเภท	อธิบาย
Port Scanning	ตรวจว่าพอร์ตไหนเปิด
IP Spoofing	ปลอม IP ปลายทาง, ซ่อนตัว
TCP Covert Channel	แอบส่งข้อมูลลับผ่านช่องทางซ่อน เช่น ICMP
IP Fragment Attack	ใช้การแยกแพ็กเกตให้หลอก firewall (เช่น Ping of Death, Tiny Fragment)
TCP Flag Attack	ส่ง flag ผิดปกติ เช่น SYN+FIN จนระบบงง
TCP Session Hijacking	แย่ง session TCP ที่ยังเปิดอยู่ (ผ่านการเดา Seq. No หรือ ARP Spoofing)
SYN Flood	ส่ง SYN เยอะๆ ทำให้ server ค้างที่รอ ACK

📌 4. Denial-of-Service (DoS/DDoS)
- DoS: โจมตีให้ระบบไม่สามารถให้บริการได้ เช่น ใช้ CPU, RAM, bandwidth จนหมด
- DDoS: ใช้หลายเครื่องพร้อมกัน (botnet) → โจมตีหนักขึ้น

ประเภทการโจมตี:
- Flooding: เช่น ICMP Flood, UDP Flood, TCP SYN Flood
- Application Attacks: เช่น HTTP flood, SIP flood, Slowloris
- Spoofing: ปลอม IP แล้วโจมตี → ตรวจกลับลำบาก
- Reflection + Amplification:
  - ปลอมว่าเป็น IP เหยื่อ ส่ง request เล็กๆ ไปหา server → server ตอบกลับขนาดใหญ่ใส่เหยื่อ
  - เช่น DNS Amplification

📌 5. เทคนิคการป้องกันการโจมตี
ก่อนการโจมตี:
- Block spoofed IP ใกล้ต้นทาง
- ปรับพารามิเตอร์ TCP/IP เช่น connection timeout
- ใช้ load balancer, CAPTCHA, mirrored server

ระหว่างโจมตี:
- ตรวจจับ pattern แปลก
- Filter packet แปลก

หลังการโจมตี:
- มีแผนรับมือ (Incident Response Plan)
- ตรวจสอบ log, หาแหล่งที่มา (traceback)
- ประสานงานกับ ISP

1. Cross-Site Scripting (XSS) [OWASP A3:2021]
XSS คือการโจมตีที่แฮกเกอร์ inject โค้ดอันตราย (เช่น JavaScript) เข้าสู่ browser ของผู้ใช้งานผ่านเว็บที่ดูเหมือนไม่มีพิษภัย
การป้องกัน:
- เข้ารหัสตัวพิเศษ <, > ด้วย htmlspecialchars() ใน PHP
- ใช้ library เช่น jsoup, OWASP ESAPI, AntiXSS
- กรอง input + encode output
-

2. Cryptographic Failure [OWASP A2:2021]
คือการใช้การเข้ารหัสผิดวิธี เช่นไม่เข้ารหัส, ใช้ algorithm เก่า, หรือเก็บรหัสผ่านไม่ปลอดภัย
- หลีกเลี่ยง DES, 3DES, RC4
- ใช้ secure random และระบบจัดการกุญแจ (Key Management)

3. SQL Injection (SQLi) [OWASP A3:2021]
คือการโจมตีฐานข้อมูลด้วยการแทรกคำสั่ง SQL ผ่าน input field
- ใช้ Prepared Statement (Parametrized Query) → แยกคำสั่งและข้อมูล

4. Security Misconfiguration [OWASP A5:2021]
การตั้งค่าที่ไม่ปลอดภัยของระบบ เช่น เปิดหน้า default, ไม่ปิด error message
- System Hardening (ลบ component ที่ไม่ใช้, ปิดฟีเจอร์ไม่จำเป็น)
- จำกัดสิทธิ์การเข้าถึง (Least Privilege)

Malicious Software
- ซอฟต์แวร์ที่ถูกแทรกเข้ามาในระบบโดยมีจุดประสงค์เพื่อ ทำลาย, ขโมยข้อมูล, หรือ รบกวนการทำงาน ของระบบ
- มี 2 ส่วนหลัก:
  - Propagation Mechanism: วิธีแพร่กระจายตัวเอง
  - Payload: สิ่งที่มัลแวร์ทำเมื่อเข้าระบบ เช่น ลบไฟล์ ขโมยข้อมูล

1. Virus
- แทรกตัวในไฟล์หรือโปรแกรม ต้องมี host

2. Worm
- โปรแกรมอิสระที่แพร่กระจายตัวเองผ่านเครือข่าย
- ไม่ต้องมี host

3. Trojan Horse
- โปรแกรมที่แสร้งว่าไม่มีอันตราย เช่น เกม, โปรแกรมฟรี
- เมื่อเปิด → ทำงานแอบแฝง เช่น ติด backdoor, ขโมยข้อมูล

4. Backdoor / Trapdoor
- ช่องทางลับที่โปรแกรมเมอร์ใส่ไว้ (บางครั้งลืมลบ)
- แฮกเกอร์สามารถใช้ช่องนี้เข้าระบบได้

5. Logic Bomb
- โค้ดที่แฝงในโปรแกรม ทำงานเมื่อเงื่อนไขตรง (เวลา, ไฟล์, ผู้ใช้)
- ใช้ทำลายระบบแบบเจาะจง

6. Rootkit
- ซ่อนตัวในระบบระดับลึก (root/admin) เพื่อควบคุมโดยไม่ถูกตรวจพบ
- มักมากับ Trojan

7. Bot / Zombie
- เครื่องคอมที่ถูกควบคุมจากระยะไกล
- ใช้ในการโจมตี DDoS แบบ botnet

8. Ransomware
- เข้ารหัสไฟล์ในเครื่อง แล้วเรียกค่าไถ่เพื่อปลดล็อก

เทคนิคการตรวจจับมัลแวร์
- Scanning: ตรวจจาก signature
- Heuristics: วิเคราะห์พฤติกรรม/ลักษณะน่าสงสัย
- Integrity checking: ตรวจความถูกต้องของไฟล์ (hash/checksum)
- Behavior blocking: ตรวจพฤติกรรมไม่พึงประสงค์
- Generic Decryption (GD): ปล่อยให้มัลแวร์ถอดรหัสตัวเองเพื่อสแกนหา

วิธีป้องกันและจัดการมัลแวร์ (Countermeasures)
การป้องกัน:
- อัปเดตระบบเสมอ
- ตั้งค่าการเข้าถึงที่เหมาะสม
- ให้ความรู้กับผู้ใช้

หากตรวจพบ:
- พยายามลบหรือแยกไวรัสออก
- กู้ไฟล์จาก backup
- หากลบไม่ได้ ต้องลบไฟล์ทิ้ง

ซอฟต์แวร์ Antivirus:
- Gen1: Signature
- Gen2: Heuristic + checksum
- Gen3: ตรวจพฤติกรรม
- Gen4: แบบผสมผสาน

Network Security
1. Perimeter Security
- ใช้ Firewall + Router บริเวณขอบเครือข่าย (edge)
- เพื่อควบคุมข้อมูลเข้า-ออก (ตาม Protocol, IP, Content ฯลฯ)

โครงสร้างป้องกัน
- DMZ (Demilitarized Zone): โซนพิเศษให้บุคคลภายนอกเข้าถึงเว็บ/อีเมล/FTP แต่ไม่สามารถเข้าเครือข่ายหลัก
- Intranet: สำหรับผู้ใช้ภายใน
- Extranet: สำหรับพันธมิตร/ลูกค้าที่เชื่อถือได้ (ไม่ใช่ public)

2. อุปกรณ์ป้องกันเครือข่าย
- Firewall: กำแพงคัดกรอง packet
- IDS (Intrusion Detection): ตรวจจับการบุกรุก
- IPS (Intrusion Prevention): ป้องกันการบุกรุก
- Spam Filter, Antivirus, Content Scanner

Firewall คืออะไร

- ซอฟต์แวร์หรือฮาร์ดแวร์ที่ทำหน้าที่กรองการรับส่งข้อมูลเข้า-ออกเครือข่าย
- ใช้เพื่อป้องกันการเข้าถึงโดยไม่ได้รับอนุญาต
- ติดตั้งที่จุดเชื่อมต่อ (perimeter) ของเครือข่าย

วัตถุประสงค์ของ Firewall

1. ให้ข้อมูลผ่าน firewall เท่านั้น
2. อนุญาตเฉพาะข้อมูลที่ถูกกำหนดไว้ใน policy
3. Firewall ต้องแข็งแรง ไม่ถูกแฮกง่าย

ข้อดี/ข้อจำกัดของ Firewall

- รวมจุดควบคุมเดียว / ตรวจสอบได้
- X ไม่สามารถป้องกันภัยจากภายใน
- X ไม่เห็น traffic ที่ bypass firewall เช่น VPN, BYOD

ชนิดของ Firewall

ประเภท	รายละเอียด
Packet Filtering (Stateless)	กรองจาก header เช่น IP, port, protocol
Stateful Inspection	จดจำสถานะ connection และอนุญาตเฉพาะ packet ที่ถูกต้องตามลำดับ
Application-Level Gateway (Proxy)	ตรวจสอบ traffic ระดับแอป เช่น HTTP/FTP
Circuit-Level Gateway	ตรวจเฉพาะ TCP session ไม่ตรวจเนื้อหา

สรุปเปรียบเทียบ (ใช้ทวนสอบได้เร็ว)

หัวข้อ	IDS	IPS
ตำแหน่ง	Passive (มองเฉยๆ)	Inline (อยู่กลางทาง)
ทำอะไรได้	ตรวจจับ + แจ้งเตือน	ตรวจจับ + ป้องกัน
ประเภท	HIDS / NIDS / Hybrid	HIPS / NIPS
ตัวอย่าง	Snort, Suricata	Snort Inline

เทคนิค	อธิบาย
Signature Detection	ตรวจจับตามรูปแบบที่รู้จัก (เช่น rule ของ SNORT)
Anomaly Detection	ตรวจพฤติกรรมแปลก เช่น ใช้งานเกินปกติ หรือโปรโตคอลเปลี่ยนแปลง

IDS/IPS

IDS (ระบบตรวจจับการบุกรุก)

- ตรวจจับพฤติกรรมผิดปกติหรือเจตนาโจมตี

ประเภท:

- HIDS: ตรวจในเครื่องเดียว (Host)
- NIDS: ตรวจ traffic ในเครือข่าย (Network)
- Distributed/Hybrid: รวม HIDS + NIDS

IPS (ระบบป้องกันการบุกรุก)

- เหมือน IDS แต่สามารถ บล็อกการโจมตีได้ทันที
- ทำงานแบบ inline (อยู่ในเส้นทาง traffic)
- ทำหน้าที่คล้าย firewall + IDS

📌 IPS ประเภท:

- Host-Based IPS: ป้องกันเฉพาะเครื่อง
- Network-Based IPS: บล็อกแพ็กเกต/เชื่อมต่อที่ผิดปกติ

7. Cloud Security Alliance (CSA)

CSA CCM (Cloud Controls Matrix)

- Framework จัดการความเสี่ยงในคลาวด์
- ช่วยประเมินและปรับปรุงมาตรการความปลอดภัย

CSA STAR

ระดับ	รายละเอียด
Level 1	Self-Assessment
Level 2	3rd-Party Certification (เช่น ISO 27001)
Level 3	Continuous Auditing (ตรวจสอบตลอดเวลา)

หลักการป้องกันแบบ Layered Physical Security

1. Deterrence – การข่มขู่/ป้องปราม (ป้ายเตือน, รั้ว, รปภ.)
2. Delaying – ถ่วงเวลาไม่ให้เข้าถึงระบบได้เร็ว (ล็อก, กำแพง, ประตูนิรภัย)
3. Detection – ตรวจจับการบุกรุก (CCTV, Motion Detector, Alarm)
4. Assessment – วิเคราะห์ความเสียหายหรือเหตุการณ์
5. Response – มีแผนตอบสนอง (ดับเพลิง, อพยพ, แจ้งเตือน)

ประเภทของภัยคุกคาม (Physical Security Threats)

1. Environmental Threats (ภัยจากสิ่งแวดล้อม)
   - ภัยธรรมชาติ: น้ำท่วม, ไฟไหม้, พายุ, แผ่นดินไหว
   - อุณหภูมิ/ความชื้นไม่เหมาะสม: ร้อนเกินไป, เย็นเกินไป, ความชื้นสูงหรือต่ำเกิน
   - ฝุ่น/แมลง/เชื้อรา: ทำให้อุปกรณ์เสียหาย
2. Technical Threats (ภัยจากเทคนิค/ไฟฟ้า)
   - ไฟตก ไฟดับ ไฟกระชาก (Surge)
   - สัญญาณรบกวนจากคลื่นแม่เหล็กไฟฟ้า (EMI)
3. Human-Caused Threats (ภัยจากมนุษย์)
   - การเข้าถึงโดยไม่ได้รับอนุญาต (Unauthorized access)
   - ขโมยอุปกรณ์หรือข้อมูล
   - การก่อวินาศกรรม (Vandalism)
   - พนักงานไม่พอใจภายในองค์กร (Insider Threat)

สรุป Identification & Authentication

คำศัพท์	ความหมาย
Identification	การบอกตัวตน (User ID, Email, MAC Address)
Authentication	การพิสูจน์ว่าเป็นเจ้าของตัวจริงของ ID
Authorization	การกำหนดสิทธิ์หลังจากผ่านการยืนยันตัวตน

สิ่งที่รู้ (Something you know)	รหัสผ่าน, PIN
สิ่งที่มี (Something you have)	บัตร, Token, Smart card
สิ่งที่เป็น (Something you are)	ลายนิ้วมือ, ใบหน้า, เสียง
สิ่งที่ทำ (Something you do)	ลายเซ็น, การพิมพ์

Multi-Factor Authentication (MFA) = ใช้มากกว่า 1 อย่างร่วมกัน เช่น รหัส + OTP

ประเภทการโจมตี	วิธีป้องกัน
Guessing, Brute-force	ใช้ password ที่ยาก, มี lockout
Shoulder Surfing	ฝึกความรู้สึกตัว, ใช้ MFA
Rainbow Table	ใช้ Salt, Hash ที่ซับซ้อน (เช่น Bcrypt)
Replay Attack	ใช้ Challenge-Response Protocol

ระบบ UNIX ใช้ salt + hash เพื่อป้องกันรหัสผ่านที่ซ้ำกัน

Token & Biometric Authentication

- **Token:** บัตรแม่เหล็ก, Smart card, USB Token → อาจถูกขโมย
- **Biometric:** Retina (แม่นยำสูงสุด), ลายนิ้วมือ, เสียง → มี false match/false reject

Remote User Authentication

- ใช้ **Challenge-Response Protocol** เพื่อป้องกันการดักฟังและ replay
- เช่น ส่ง random number (nonce) → ให้ client ตอบกลับด้วยค่าที่ hash ไว้

Email Protocols

Protocol	หน้าที่
SMTP	ส่งอีเมล (UA → MTA)
POP3	โหลดอีเมลมาเก็บที่ client
IMAP	อ่านอีเมลจาก server หลายอุปกรณ์ได้

จุดอ่อนของ Email

ไม่มีการเข้ารหัสโดยค่าเริ่มต้น → อาจถูกดักอ่านได้

ใช้ SSL/TLS ได้ แต่ไม่ใช่ End-to-End

Phishing คือการหลอกลวงทางอีเมล เช่น หลอกให้กดลิงก์ปลอม

บริการ	อธิบาย
Confidentiality	ป้องกันไม่ให้คนอื่นอ่าน
Authentication	รู้ว่าใครเป็นผู้ส่ง
Integrity	ตรวจว่าไม่ได้ถูกแก้ไข
Non-repudiation	ปฏิเสธไม่ได้ว่าตนเป็นผู้ส่ง

PGP (Pretty Good Privacy)

ใช้ RSA + AES/3DES + SHA-1

มี 5 ขั้นตอน: Authentication, Confidentiality, Compression, Email compatibility, Segmentation

ใช้ Key Ring สำหรับเก็บ Public/Private Keys

ใช้ Radix-64 เข้ารหัสเพื่อส่งในอีเมล

S/MIME (Secure MIME)

- พัฒนาโดย RSA Security
- เสริมความปลอดภัยให้ MIME (ใช้กับอีเมลทั่วไปได้)
- ใช้ X.509 Certificates, support:
  - Signed Data → สำหรับ Integrity
  - Enveloped Data → สำหรับ Confidentiality
  - Digested Data → Hash อย่างเดียว
  - Authenticated Data → มี MAC

สรุป Cloud Security (ความปลอดภัยบนคลาวด์)

1. พื้นฐานของ Cloud Computing

ประเภท Cloud	ความปลอดภัย	ความคุ้มค่า
Private	สูงสุด	แพงที่สุด
Public	ปานกลาง	ถูกที่สุด
Hybrid	ผสมดีทั้งคู่	ปานกลาง
Community	ใช้ร่วมกัน	ปานกลาง

2. แนวคิดความปลอดภัยบน Cloud (จาก NIST SP 800-144)

ด้าน	อธิบาย
Governance	นโยบาย/กระบวนการควบคุมภายในองค์กร
Compliance	ปฏิบัติตามกฎหมาย เช่น PDPA, GDPR
Trust	มั่นใจว่า provider ดูแลความปลอดภัยได้
Architecture	เข้าใจโครงสร้างของระบบคลาวด์
Identity & Access Management (IAM)	ควบคุมการเข้าถึงและสิทธิของผู้ใช้
Software Isolation	Virtualization แยกผู้ใช้ออกจากกัน
Data Protection	ป้องกันข้อมูลขณะพัก, ส่ง, ใช้งาน
Availability	ความต่อเนื่องของระบบ, สำรองข้อมูล
Incident Response	การตอบสนองเหตุการณ์ฉุกเฉิน

3. Shared Responsibility Model

- **Provider:** จัดการ hardware, hypervisor, physical security
- **Customer:** จัดการข้อมูล, การเข้าถึง, app-level security
- ยิ่งไปทาง SaaS → ความรับผิดชอบลูกค้าน้อยลง

4. ความเสี่ยงสำคัญของ Cloud

ตามรูปแบบ Deployment Models

- **Private Cloud:** ควบคุมได้ดี แต่ยังมีภัยจากพนักงาน, ภัยธรรมชาติ
- **Public Cloud:** เสี่ยงเรื่อง vendor lock-in, multitenancy, legal
- **Community Cloud:** ความเสี่ยงกระจาย, ทุก node คือจุดอ่อน

ตาม Service Models

- **IaaS:** ลูกค้าคุมมาก → ต้องมี skill เยอะ
- **PaaS:** เสี่ยงเรื่อง compatibility, backdoor
- **SaaS:** เสี่ยงจาก API, Web app, data lock-in

5. Top 11 Cloud Threats (2024)

1. **Misconfiguration** – ตั้งค่าผิด เช่น S3 เปิด public
2. **IAM** – ตั้งสิทธิ์เกินจำเป็น, ไม่ใช้ least privilege
3. **Insecure APIs** – ไม่มี auth, input validation ไม่ดี
4. **No Cloud Security Strategy** – ขาดการวางแผนล่วงหน้า
5. **Insecure Third-Party Resources** – ใช้โค้ดภายนอกที่ไม่ปลอดภัย
6. **Insecure Software Development** – ช่องโหว่จากการเขียนโปรแกรม
7. **Accidental Data Disclosure** – ข้อมูลรั่วจาก repo public
8. **System Vulnerabilities** – ช่องโหว่จาก OS, library
9. **Limited Visibility** – ไม่เห็นว่าใครใช้ cloud ทำอะไร (Shadow IT)
10. **Unauthenticated Sharing** – แชร์ VM, storage โดยไม่กำหนดสิทธิ์
11. **Advanced Persistent Threats (APT)** – กลุ่มโจมตีระดับสูง (เช่น ransomware)

6. Cloud Security Approaches

Data Protection

- Data at Rest / In Transit / In Use → ต้องเข้ารหัสและควบคุมการเข้าถึง
- Multi-tenant vs Multi-instance → ควรเลือกตามความปลอดภัยที่ต้องการ

Security as a Service (SECaaS)

บริการ	หน้าที่
IAM	จัดการสิทธิการเข้าถึง
DLP	ป้องกันข้อมูลรั่วไหล
Web & Email Security	ป้องกัน phishing, malware
Intrusion Detection/Prevention	ตรวจจับ/บล็อกการโจมตี