1. หากได้รับรายงานว่ามีการโจมตีระบบเครือข่ายของ บริษัทผ่านทางอินเทอร์เน็ต จากผู้ไม่ประสงค์ดี ซึ่ง บางอย่างมีการบล็อกการเข้าถึงบางเว็บไซต์ และการส่ง อีเมล์สแปมจำนวนมาก จะวิเคราะห์และป้องกันการ โจมตีนี้อย่างไร จะต้องทำการวิเคราะห์และป้องกันการโจมตีดังนี้: 1. วิเคราะห์ชนิดของการโจมตี: จะต้องวิเคราะห์และระบุว่าการโจมตีเป็นประเภทใด เช่น การโจมตีแบบ DDoS (Distributed Denial of Service) หรือการโจมตีแบบเจาะระบบ (System Hacking) เป็นต้น เพื่อทำการป้องกันด้วยวิธีที่เหมาะสม 2. ตรวจสอบและปรับปรุงระบบความปลอดภัย: จะต้องตรวจสอบและปรับปรุงระบบความปลอดภัยของเครือข่ายอย่างสม่ำเสมอ เช่น การอัปเดต ซอฟต์แวร์หรือฮาร์ดแวร์เพื่อเติมเต็มช่องโหว่ (Vulnerability) ที่อาจเป็นประเด็นในการโจมตี 3. การควบคุมการเข้าถึง: จะต้องทำการตั้งค่าการเข้าถึงของผู้ใช้งานและระบบ เพื่อป้องกันการเข้าถึงแหล่งข้อมูลที่มากับอุปกรณ์ที่ใช้เชื่อมต่อกับ เครือข่าย เช่น อุปกรณ์ IoT (Internet of Things) ที่มีความปลอดภัยน้อย อาจเป็นทางเข้าให้กับผไมประสงคด 4. การตดตงระบบ IDS/IPS: ควรตดตงระบบ IDS (Intrusion Detection System) และ IPS (Intrusion Prevention System) เพอตรวจสอบ และปองกนการเขาถงไมมอำนาจ หรอการโจมตบนเครอขาย 5. การเขยนนโยบายความปลอดภย: ควรเขยนนโยบายความปลอดภยเพอเตรยมตวและปองกนการโจมตในอนาคต โดยควรรวบรวมแนวทางการ ปองกนและวธการเฝาระวง รวมถงการดำเนนการในกรณเกดการโจมต 6. การสอนพนกงาน: ควรสอนพนกงานทกระดบในองคกรเกยวกบความสำคญของความปลอดภยเครอขาย รวมถงการใชรหสผานทแขงแกรงและ การเปดเผยขอมลทเปนความลบอยางระมดระวง

᲼᲼ — 06/05/2023 10:13

นอกจากการทำ Incident Response Plan แล้ว สามารถทำอะไรเพิ่มเติมเพื่อป้องกันเหตุฉุกเฉินได้บ้าง นอกจากการทำ Incident Response Plan แล้ว ยังมีกลไกต่างๆที่องค์กรสามารถใช้เพื่อป้องกันเหตุการณ์ร้ายแรง ได้แก่ 1. การอัพเดทระบบ: อัพเดทระบบขององค์กรเป็นปัจจัยสำคัญในการป้องกันการโจมตี โดยทำการอัพเดทระบบประจำเวลาจะช่วยลดโอกาสของ การโจมตีที่ใช้ช่องโหว่ในระบบ ซึ่งทางผู้ดูแลระบบควรให้ความสำคัญกับการอัพเดทระบบและแก้ไขช่องโหว่ในระบบอย่างรวดเร็ว 2. การตรวจสอบและตรวจจับ: องค์กรควรติดตั้งโปรแกรมที่ช่วยตรวจสอบและตรวจจับการแทรกแซง (intrusion detection/prevention system) เพื่อช่วยตรวจจับและป้องกันการโจมตีได้ทันที รวมถึงการใช้โปรแกรมป้องกันไวรัสและโปรแกรมป้องกันการเข้าถึงไม่หมายความจริง (anti-malware/anti-spyware) ซึ่งช่วยป้องกันไม่ให้โปรแกรมอันตรายเข้ามาในระบบ 3. การเข้ารหัสข้อมูล: การเข้ารหัสข้อมูล (encryption) เป็นวิธีการป้องกันข้อมูลที่สำคัญและเป็นความลับไม่ให้ถูกดักรับข้อมูล ทางองค์กรควรใช้ การเข้ารหัสข้อมูลเป็นประจำเมื่อส่งข้อมูลข้ามเครือข่ายหรือออกสู่สาธารณะ 4. การทำการสำรองข้อมูล (backup): การทำการสำรองข้อมลเปนวธการทชวยลดความเสยงในกรณทขอมลสญหาย (มีการแก้ไข)

1. [10:16]

นอกจากมาตรฐาน ISO 27001 มีมาตรฐานอะไรที่ สามารถสร้างความน่าเชื่อถือให้กับองค์กรได้บ้าง นอกเหนือจากมาตรฐาน ISO 27001 มีมาตรฐานอื่นๆ ที่สามารถสร้างความเชื่อถือได้ เช่น 1. NIST Cybersecurity Framework: เป็นแนวทางในการสร้างและดำเนินการด้านความมั่นคงปลอดภัยสำหรับองค์กร โดยมุ่งเน้นการป้องกันการ โจมตีและการตอบสนองต่อการโจมตีโดยใช้วิธีการเรียงลำดับ จัดกลุ่ม และอธิบายแนวทางปฏิบัติที่ดีที่สุด 2. PCI DSS: เป็นมาตรฐานสำหรับการป้องกันการขโมยข้อมูลการเงินในระบบการช าระเงิน มีจุดประสงค์เพื่อป้องกันการขโมยข้อมูลบัตรเครดิต ป้องกันการสอบสวนและลดความเสี่ยงในการสูญเสียรายได้ 3. CIS Controls: เป็นแนวทางควบคุมการทำความมั่นคงปลอดภัยที่สามารถป้องกันการโจมตีได้มากที่สุด โดยประกอบด้วย 20 ขั้นตอนในการ ปรับปรุงการดำเนินงานของระบบคอมพิวเตอร์เพื่อเพิ่มความมั่นคงปลอดภัย การใช้มาตรฐานต่างๆ จะช่วยให้องค์กรเข้าใจถึงความเสี่ยงและอุปสรรคที่อาจเกิดขึ้นในการดำเนินงาน และช่วยเพิ่มความมั่นคงปลอดภัยของระบบและ ข้อมูลในองค์กร

infrastructure mode

Client / server หรือ Infrastructure mode เป็นลักษณะการรับส่งข้อมูลโดยอาศัยAccess Point (AP)

หรือเรียกว่า “Hot spot” ทำหน้าที่เป็นสะพานเชื่อมต่อระหว่างระบบเครือข่ายแบบใช้สายกับเครื่องคอมพิวเตอร์ลูกข่าย (client)

โดยจะกระจายสัญญาณคลื่นวิทยุเพื่อ รับ-ส่งข้อมูลเป็นรัศมีโดยรอบ เครื่องคอมพิวเตอร์ที่อยู่ในรัศมีของ AP จะกลายเป็น เครือข่ายกลุ่มเดียวกันทันที

โดยเครื่องคอมพิวเตอร์ จะสามารถติดต่อกัน หรือติดต่อกับ Serverเพื่อแลกเปลี่ยนและค้นหาข้อมูลได้ โดยต้องติดต่อผ่านAP

เท่านั้น ซึ่ง AP 1 จุด สามารถให้บริการเครื่องลูกข่ายได้ถึง 15-50อุปกรณ์ ของเครื่องลูกข่าย เหมาะสำหรับการนำไปขยายเครือข่ายหรือใช้ร่วมกับระบบเครือข่าย

แบบใช้สายเดิมในออฟฟิต,ห้องสมุด หรือในห้องประชุม เพื่อเพิ่มประสิทธิภาพในการทำงานให้มากขึ้น

ad hoc mode

Peer to Peer (ad-hoc mode)

เป็นรูปแบบการเชื่อมต่อแลนไร้สายที่มีลักษณะการเชื่อมต่อแบบโครงข่ายโดยตรงระหว่างเครื่องคอมพิวเตอร์จำนวน 2 เครื่องหรือมากกว่านั้น

โดยเครื่องคอมพิวเตอร์แต่ละเครื่องจะมีความเท่าเทียมกัน การเชื่อมต่อด้วยวิธีนี้จะเหมาะกับเครือข่ายขนาดเล็กที่มีโหนดเชื่อมต่อจำนวนไม่มาก

SSID ย่อมาจาก (Service Set Identifier) ซึ่งเป็นชื่อของ WIFI ถ้ายังมองภาพไม่ออก ก็ให้หยิบมือถือขึ้นมา แล้วเปิดให้ WIFI ทำงาน เราจะเห็นว่ารอบ ๆ ตัวเรานั้นมีชื่อ WIFI อะไรบ้างที่ปรากฎขึ้นมา ชื่อตรงนั้นคือสิ่งที่เรียกว่า SSID นั่นเอง ชื่อเหล่านี้เราสามารถตั้งได้ตามความต้องการ หากที่บ้านของเอาติดตั้งอินเตอร์เน็ต และมีอุปกรณ์กระจายสัญญาณ WIFI ซึ่งการจัดตั้งชื่อ SSID ได้นั้น จะต้องเข้าสู่ระบบของ Router ผ่านทางโปรแกรม Web browser โดยเมื่อเปิดโปรแกรม Web Browser ขึ้นมาแล้วค่าเริ่มต้นของหมายเลข IP Address ที่จะใช้สำหรับเข้าสู่การตั้งค่า SSID ก็คือ 192.168.1.1 หลังจากนั้นจะมีหน้าต่างสำหรับใส่ Username และ Password ขึ้นมา และค่าเริ่มต้นของ Username และ Password นั้นสามารถดูได้จากคู่มือ หรือใต้เครื่อง Router หรือ Access Point

เมื่อเข้าสู่ระบบได้แล้ว ให้มองหาเมนู Network > WLAN Radio 2.4G > SSID Setting และเข้าไปตั้งชื่อที่ต้องการเป็นภาษาอังกฤษในช่อง SSID Name สำหรับการตั้งชื่อนั้น แนะนำให้ใช้ชื่อง่าย ๆ ที่เป็นเอกลักษณ์ที่บ่งบอกความเป็นตัวของเรา การทำการตั้งชื่อควรตรวจเช็คให้ดีว่าชื่อ SSID หรือชื่อ WIFI ที่อยู่ใกล้เคียงนั้นมีชื่อที่เราต้องการแล้วหรือยัง เพราะไม่ควรตั้งชื่อ WIFI หรือ SSID ซ้ำกัน เพราะจะทำให้มีปัญหาในการเชื่อมต่อได้

**SSID Broadcast คืออะไร?**

นี่คือฟังก์ชั่นที่ดำเนินการโดยจุดเชื่อมต่อที่ส่งชื่อของมันซึ่งสถานีไร้สายที่ค้นหาการเชื่อมต่อเครือข่ายสามารถค้นพบ 'มันได้ นี่คือสิ่งที่ช่วยให้โปรแกรมตัวจัดการไคลเอ็นต์ของอแด็ปเตอร์ไร้สายของคุณหรือซอฟต์แวร์ไร้สายในตัวของ Windows XP สามารถแสดงรายการจุดเข้าใช้งานในระยะ

ความสามารถในการปิดใช้งานการออกอากาศ SSID โดยพื้นฐานแล้วจะทำให้ Access Point ของคุณมองไม่เห็นเว้นแต่ว่าไคลเอนต์ไร้สายจะรู้จัก SSID อยู่แล้วหรือกำลังใช้เครื่องมือที่มอนิเตอร์หรือ 'sniff' ทราฟฟิกจากไคลเอนต์ที่เกี่ยวข้องของ AP

[WEP, WPA, WPA2 และ WPA3 มาตรฐานความปลอดภัย Wi-Fi คืออะไร ? แตกต่างกันอย่างไร ? (thaiware.com)](https://tips.thaiware.com/1577.html)

Open System Authentication is a type of authentication method used in wireless networks. It is the simplest form of authentication and provides no real security measures. In Open System Authentication, any device can connect to the wireless network without the need for authentication credentials, such as a password or encryption key.

Here's how Open System Authentication works:

1. Device Discovery: The wireless network broadcasts its Service Set Identifier (SSID) to inform nearby devices of its presence.
2. Association Request: A device that wants to connect to the wireless network sends an association request to the access point (AP) or wireless router.
3. Access Point Response: The access point, in an open system network, accepts the association request without requiring any authentication. It sends an association response to the device, allowing it to join the network.
4. Network Access: After receiving the association response, the device is granted access to the wireless network and can communicate with other devices within the network range.

Open System Authentication is considered "open" because it does not authenticate or encrypt any data exchanged between the wireless devices and the network. This means that any device within range of the network can connect to it and potentially eavesdrop on or intercept network traffic.

As a result, Open System Authentication is not recommended for securing sensitive information or protecting against unauthorized access. It is often used in public Wi-Fi networks or scenarios where convenience and ease of connectivity take precedence over security.

To enhance the security of wireless networks, it is advisable to use stronger authentication methods such as WPA2 (Wi-Fi Protected Access 2) or WPA3, which employ encryption and more robust authentication mechanisms to ensure secure communication between devices and the network.

Key authentication, also known as public key authentication, is a method of securely verifying the identity of a user or entity in computer systems. It is commonly used in secure communication protocols such as SSH (Secure Shell) to authenticate remote access.

In key authentication, a user has a pair of cryptographic keys: a public key and a private key. The public key is shared with other parties or stored on servers, while the private key is kept confidential and securely stored on the user's device.

Here's how key authentication works:

1. Key Generation: The user generates a key pair consisting of a public key and a corresponding private key. The private key remains on the user's device, while the public key is shared with the remote server or the intended party.
2. Public Key Distribution: The user's public key is distributed to the servers or parties that need to authenticate the user's identity. This can be done by manually sharing the public key or through a key distribution infrastructure.
3. Requesting Authentication: When the user wants to access a remote system (e.g., SSH into a server), the remote system requests the user to authenticate.
4. Client Authentication: The client (user) sends a message to the server, typically including the user's identification and a cryptographic challenge.
5. Signing the Challenge: The client signs the challenge using its private key, creating a digital signature.
6. Verification: The server retrieves the client's public key and uses it to verify the digital signature. If the signature is valid, the server can be confident that the client possesses the corresponding private key and is therefore the legitimate user.
7. Access Granted: If the verification is successful, the server grants access to the client, allowing the user to log in or perform the requested actions.

Key authentication provides a higher level of security compared to traditional password-based authentication. As the private key is never shared or transmitted, it is less susceptible to interception or brute-force attacks. Additionally, even if the public key is compromised, it does not jeopardize the security of the private key.

It's important to note that key authentication requires careful management and protection of the private key to prevent unauthorized access. Users should ensure their private keys are stored securely and take precautions to prevent unauthorized access to their devices or key storage locations

no clean

\*\*\*

Broken Access Control เป็นช่องโหว่ด้านความปลอดภัยที่เกิดขึ้นเมื่อแอปพลิเคชันหรือระบบไม่จำกัดการเข้าถึงทรัพยากรหรือฟังก์ชันของตนอย่างถูกต้อง

ช่องโหว่นี้ทำให้ผู้ใช้ที่ไม่ได้รับอนุญาตสามารถดำเนินการหรือเข้าถึงข้อมูลที่ไม่ควรได้เข้าถึงได้

มีหลายวิธีที่ Broken Access Control สามารถเกิดขึ้นในแอปพลิเคชันหรือระบบได้ ตัวอย่างขั้นต้นเช่น

\*\*\*\*\*

ขาดการตรวจสอบสิทธิ์การเข้าใช้: แอปพลิเคชันหรือระบบไม่ต้องการผู้ใช้ยืนยันตัวตนก่อนที่จะอนุญาตให้เข้าถึงทรัพยากรหรือฟังก์ชัน

ตัวอย่างเช่น พิจารณาแอปพลิเคชันเว็บที่อนุญาตผู้ใช้ดูข้อมูลบัญชีของตนเอง เช่น ประวัติการซื้อและข้อมูลการเรียกเก็บเงิน แต่แอปพลิเคชันไม่ต้องการผู้ใช้เข้าสู่ระบบหรือให้ข้อมูลประจำตัวใดๆก่อนที่จะให้เข้าถึงข้อมูลดังกล่าว

ผู้โจมตีสามารถใช้ช่องโหว่นี้โดยการเข้าถึง URL ของหน้าข้อมูลบัญชีโดยตรงหรือโดยการแก้ไขฟิลด์อินพุตเพื่อเข้าถึงข้อมูลบัญชีผู้ใช้อื่น ๆ ซึ่งสามารถทำให้ผู้โจมตีเข้าถึงข้อมูลที่สำคัญหรือดำเนินการที่ไม่ได้รับอนุญาตได้

เช่น เปลี่ยนการตั้งค่าบัญชีหรือทำการซื้อสินค้า

\*\*\*\*

การตรวจสอบสิทธิ์ไม่เพียงพอ: แอปพลิเคชันหรือระบบตรวจสอบสิทธิ์การเข้าถึงของผู้ใช้ แต่ไม่มีความแข็งแกร่งพอหรือไม่ครอบคลุมทุกสถานการณ์

ตัวอย่างเช่น พิจารณาโปรแกรมเว็บที่อนุญาตให้ผู้ใช้อัพเดทข้อมูลโปรไฟล์ของตนเอง เช่นที่อีเมลและรหัสผ่าน โปรแกรมตรวจสอบว่าผู้ใช้มีสิทธิ์การเข้าถึงและการอัพเดทโปรไฟล์ แต่โปรแกรมเช็คเพียงว่าผู้ใช้มีสิทธิ์เข้าถึงและอัพเดทโปรไฟล์ของตนเองเท่านั้น

แต่ไม่ตรวจสอบว่าผู้ใช้มีสิทธิ์เข้าถึงและอัพเดทโปรไฟล์ของผู้ใช้อื่น

ผู้โจมตีสามารถใช้ช่องโหว่นี้โดยการแก้ไข URL หรือข้อมูล input เพื่อเข้าถึงหน้าอัพเดทโปรไฟล์ของผู้ใช้อื่น และเปลี่ยนอีเมลและรหัสผ่านได้ ซึ่งจะทำให้ผู้โจมตีสามารถเข้าถึงข้อมูลที่ละเอียดอ่อนหรือดำเนินการโดยไม่ได้รับอนุญาต

\*\*\*\*\*

การอ้างอิงออบเจกต์โดยตรง: แอปพลิเคชันหรือระบบใช้ข้อมูลที่ผู้ใช้ส่งเข้ามาเพื่อเข้าถึงทรัพยากรโดยตรงโดยไม่มีการตรวจสอบหรืออนุญาตให้เข้าถึง

กิดขึ้นเมื่อแอปพลิเคชันเปิดเผยอ้างอิงถึงอ็อบเจกต์ภายในของการปฏิบัติงาน เช่น ไฟล์ บันทึกฐานข้อมูล หรือทรัพยากร ในรูปแบบพารามิเตอร์หรือคีย์ใน URL

หรือฟอร์มฟิลด์ ผู้โจมตีสามารถจัดการอ้างอิงเหล่านี้เพื่อเข้าถึงทรัพยากรที่ไม่ได้รับอนุญาตหรือดำเนินการที่ไม่ได้รับอนุญาต

ตัวอย่างเช่น พิจารณาแอปพลิเคชันเว็บที่อนุญาตให้ผู้ใช้ดูประวัติการซื้อของตนเองโดยให้หมายเลข ID การซื้อใน URL เช่น https://example.com/purchases?id=123

หากแอปพลิเคชันไม่ตรวจสอบว่าผู้ใช้มีสิทธิ์เข้าถึงการซื้อด้วยหมายเลข ID 123

อย่างเหมาะสม ผู้โจมตีสามารถเปลี่ยนหมายเลข ID ใน URL เพื่อเข้าถึงประวัติการซื้อของผู้ใช้อื่นๆ รวมถึงข้อมูลที่สำคัญเช่นที่อยู่สำหรับวางบิลและการส่งสินค้า

\*\*\*\*

อ้างอิงออบเจกต์โดยไม่ปลอดภัย: แอปพลิเคชันหรือระบบเปิดเผยการอ้างอิงออบเจกต์ภายในที่สามารถถูกแฮ็กโดยผู้โจมตีเพื่อเข้าถึงทรัพยากรโดยไม่ได้รับอนุญาต

เกิดขึ้นเมื่อแอปพลิเคชันเปิดเผยการอ้างอิงของวัตถุภายในที่อาจถูกโจมตีโดยผู้โจมตีที่ทำให้พวกเขาสามารถเข้าถึงทรัพยากรหรือกระทำการไม่ได้รับอนุญาต

ตัวอย่างเช่น พิจารณาแอปพลิเคชันเว็บที่อนุญาตให้ผู้ใช้ดูไฟล์ของตนเองโดยให้รหัสไอดีของไฟล์ใน URL เช่น https://example.com/files?id=123

ถ้าแอปพลิเคชันไม่ตรวจสอบให้แน่ใจว่าผู้ใช้มีสิทธิ์เข้าถึงไฟล์ด้วยรหัสไอดีหมายเลข 123 ผู้โจมตีสามารถเปลี่ยนรหัสไอดีใน URL เพื่อเข้าถึงไฟล์ของผู้ใช้อื่นได้

ต่อไปนี้สมมติว่าแอปพลิเคชันใช้หมายเลข ID ลำดับต่อเนื่องเป็น ID ของไฟล์ เช่น 1, 2, 3 และเป็นต้น ผู้โจมตีสามารถเดาหมายเลข ID เพื่อเข้าถึงไฟล์ของผู้ใช้อื่นได้ โดยไม่ต้องมีความรู้

เกี่ยวกับการดำเนินงานภายในของแอปพลิเคชัน นี่เป็นตัวอย่างของการอ้างอิงวัตถุโดยไม่ปลอดภัย

\*\*\*\*

Privilege escalation เป็นกระบวนการที่แอปพลิเคชันหรือระบบอนุญาตให้ผู้ใช้เพิ่มสิทธิ์ของตนเหนือระดับที่ได้รับอนุญาต ซึ่งสามารถทำให้ผู้ใช้เข้าถึงทรัพยากรที่ไม่ควรได้เข้าถึง

การเพิ่มสิทธิ์เกิดขึ้นเมื่อแอปพลิเคชันหรือระบบอนุญาตผู้ใช้เข้าถึงทรัพยากรหรือดำเนินการเกินระดับการเข้าถึงที่ได้รับอนุญาต สาเหตุของปัญหานี้สามารถเกิดขึ้นได้หลายวิธี เช่น

การประกอบด้วยช่องโหว่ในแอปพลิเคชันหรือระบบ: ผู้โจมตีสามารถใช้ช่องโหว่เช่น buffer overflows หรือ SQL injection เพื่อเข้าถึงการดูแลระบบหรือสิทธิ์พิเศษบนระบบ

การใช้ข้อมูลการเข้าถึงที่ถูกตั้งค่าตามค่าเริ่มต้นหรือค่าผิด: หากแอปพลิเคชันหรือระบบใช้ข้อมูลการเข้าถึงที่ถูกตั้งค่าตามค่าเริ่มต้นหรือค่าผิด ผู้โจมตีสามารถใช้วิธีการแบบบังคับการทำงานหรือ

ยังไม่ได้เข้ารหัสข้อมูลเพื่อเข้าถึงบัญชีผู้ดูแลระบบได้

การตั้งค่าผิดพลาด: การตั้งค่าผิดพลาดในแอปพลิเคชันหรือระบบสามารถทำให้ผู้โจมตีเข้าถึงสิทธิ์การดูแลระบบโดยการใช้การตั้งค่าที่ไม่แข็งแกร่งหรือไม่เหมาะสม

ตัวอย่างของการเพิ่มสิทธิ์คือเมื่อผู้ใช้ที่มีสิทธิ์จำกัดในแอปพลิเคชันเว็บสามารถแก้ไข URL หรือช่องข้อมูลเพื่อเข้าถึงฟังก์ชันการดูแลระบบได้ ตัวอย่างเช่น ในกรณีที่แอปพลิเค

ตัวอย่างของ Privilege escalation คือเมื่อผู้ใช้งานที่มีสิทธิ์จำกัดในแอปพลิเคชันเว็บสามารถแก้ไข URL หรือช่องกรอกข้อมูลเพื่อเข้าถึงฟังก์ชันการดูแลระบบได้

ตัวอย่างเช่นเว็บแอปพลิเคชันที่อนุญาตให้ผู้ใช้งานปกติเปลี่ยนแปลงข้อมูลโปรไฟล์ของตนเอง เช่นที่อยู่อีเมลของตนเอง หากแอปพลิเคชันไม่ตรวจสอบข้อมูลผู้ใช้งานอย่างเหมาะสมและ

ผู้โจมตีสามารถฝังโค้ดอันตรายในช่องกรอกข้อมูลอีเมล จะสามารถเข้าถึงฟังก์ชันการดูแลระบบและเข้าถึงข้อมูลที่เป็นความลับได้

(Cryptographic failures) เป็นเหตุการณ์ที่เกิดขึ้นเมื่อโปรโตคอลการเข้ารหัส อัลกอริทึม หรือระบบการเข้ารหัสไม่สามารถให้ความมั่นคงปลอดภัยและความลับตามที่ต้องการได้

การเข้ารหัสเป็นส่วนสำคัญของการคอมพิวเตอร์รุ่นใหม่ เนื่องจากใช้ในการรักษาความปลอดภัยของข้อมูลที่ละเอียดอ่อน เช่น การทำธุรกรรมทางการเงิน

ข้อมูลส่วนบุคคล และการสื่อสารทางทหาร อย่างไรก็ตาม ความล้มเหลวในด้านการเข้ารหัสสามารถทำให้เกิดการละเมิดความปลอดภัยของข้อมูล การเข้าถึงที่ไม่ได้รับอนุญาต และปัญหาความปลอดภัยอื่นๆ

สาเหตุของCryptographic failuresเหล่านี้อาจเกิดขึ้นได้จากหลายปัจจัย เช่น การจัดการคีย์ที่ไม่ดี หรือช่องโหว่ในซอฟต์แวร์หรือฮาร์ดแวร์ที่ใช้เพื่อดำเนินการเข้ารหัส

บางตัวอย่างของความล้มเหลวของการเข้ารหัสได้แก่:

Weak or predictable keys:ระบบการเข้ารหัสต้องการคีย์เพื่อทำการเข้ารหัสและถอดรหัสข้อมูล หากคีย์เหล่านี้อ่อนหรือสามารถทำนายได้ ผู้โจมตีอาจสามารถถอดรหัสข้อมูลได้ง่ายดาย

โดยการเข้ารหัสข้อมูล (cryptography) ความมั่นคงของข้อมูลที่ถูกเข้ารหัสขึ้นอยู่กับความแข็งแกร่งและความสุ่มของkeyในการเข้ารหัส (encryption key) ที่ใช้งานอยู่ keyการเข้ารหัสที่มีความอ่อนแอหรือเป็นแบบสามารถทายได้ง่ายๆ จะทำให้ผู้บุกรุกสามารถถอดรหัสข้อมูลได้อย่างง่ายดาย

Brute-force attacks: การโจมตีด้วยวิธีการBrute-force attacksเป็นการลองใช้คีย์ทุกกรณีจนกว่าจะพบคีย์ที่ถูกต้อง ระบบการเข้ารหัสที่มีคีย์อ่อนอาจเสี่ยงต่อการโจมตีชนิดนี้

การโจมตีแบบ Man-in-the-middle: การโจมตีแบบ Man-in-the-middle เป็นการแอบฟังการสื่อสารระหว่างสองฝ่ายและแก้ไขข้อมูลในกรณีที่

ระบบการเข้ารหัสไม่ได้ออกแบบให้สามารถตรวจจับหรือป้องกันการโจมตีชนิดนี้ ผู้โจมตีสามารถอ่านหรือแก้ไขข้อมูลที่ถูกเข้ารหัสได้

ตัวอย่างเช่นหากใช้วิธีการเข้ารหัสที่อ่อนแอหรือบกพร่องเพื่อป้องกันข้อมูลการเข้าสู่ระบบของผู้ใช้ ผู้โจมตีอาจสามารถถอดรหัสข้อมูลการเข้าสู่ระบบได้ง่าย ๆ และเข้าถึงระบบธนาคารออนไลน์ได้โดยไม่ได้รับอนุญาต

อย่างเช่นเดียวกันหากจัดการและเก็บรักษากุญแจการเข้ารหัสไม่ได้ดีอาจทำให้โจมตีเข้าถึงและถอดรหัสข้อมูลที่สำคัญ

อีกตัวอย่างของความล้มเหลวของการใช้การเข้ารหัสในเว็บแอปพลิเคชันคือการใช้โปรโตคอลการเข้ารหัสที่ไม่ปลอดภัย ตัวอย่างเช่นหากโปรโตคอล SSL/TLS

ที่ใช้รักษาความปลอดภัยของการสื่อสารระหว่างเว็บแอปพลิเคชันและเบราว์เซอร์ของผู้ใช้ไม่ได้กำหนดค่าไว้อย่างถูกต้อง อาจเป็นแหล่งที่มีช่องโหว่เพื่อให้ผู้โจมตีสามารถดักจับและอ่านข้อมูลที่ละเอียดอ่อน

ที่ถูกส่งผ่านระหว่างเบราว์เซอร์ของผู้ใช้และเว็บแอปพลิเคชัน

```

no clean

A03:2021-Injection

เป็นหมวดหมู่ของความเสี่ยงด้านความปลอดภัยที่อ้างอิงถึงช่องโหว่ที่อนุญาตให้ผู้โจมตีฝังรหัสที่เป็นอันตรายหรือข้อมูลลงในแอปพลิเคชัน การโจมตีแบบฝังรหัสเป็นช่องโหว่ที่พบได้บ่อยและสามารถใช้ประโยชน์ได้ในหลายวิธี เช่น SQL injection, NoSQL injection, command injection, และอื่น ๆ

การโจมตีแบบฝังรหัสเกิดขึ้นเมื่อผู้โจมตีสามารถป้อนข้อมูลที่เป็นอันตรายเข้าสู่ช่องอินพุตหรือพารามิเตอร์ของแอปพลิเคชันได้ สาเหตุทำให้แอปพลิเคชันดำเนินการคำสั่งที่ไม่ได้ตั้งใจหรืออนุญาตการเข้าถึงข้อมูลโดยไม่ได้รับอนุญาต

Insecure design หมายถึงการออกแบบซอฟต์แวร์ ฮาร์ดแวร์ หรือระบบที่มีช่องโหว่ทางความปลอดภัย ความไม่ปลอดภัยนี้เกิดจากขาดการให้ความสำคัญกับปัญหาความปลอดภัย ในระหว่างขั้นตอนการออกแบบ หรือการละเว้นการปฏิบัติตามหลักการเขียนโปรแกรมที่ปลอดภัย การออกแบบที่ไม่ปลอดภัยสามารถนำไปสู่ปัญหาความปลอดภัยต่าง ๆ เช่นการเข้าถึงที่ไม่ได้รับอนุญาต การแฝงตัวเข้าสู่ระบบ เป็นต้น ตัวอย่างหนึ่งของการออกแบบที่ไม่ปลอดภัยคือการใช้รหัสผ่านเริ่มต้นหรือรหัสผ่านที่ไม่แข็งแรงในอุปกรณ์หรือระบบต่าง ๆ นี้ทำให้ผู้โจมตีสามารถเข้าถึงระบบและขโมยหรือแก้ไขข้อมูลที่ละเอียดอ่อนได้อย่างง่ายดาย อีกตัวอย่างคือขาดการตรวจสอบข้อมูลที่นำเข้า ซึ่งอาจนำไปสู่การโจมตีด้วยการแทรกโค้ดหรือช่องโหว่ในการจัดเก็บข้อมูล ตัวอย่างเช่น ของการออกแบบที่ไม่ปลอดภัยในเว็บแอปพลิเคชันคือการใช้โปรโตคอลการสื่อสารที่ไม่ปลอดภัย เช่น HTTP แทน HTTPS ซึ่งอาจทำให้ผโจมตสามารถแอบแฝงและแกไขขอมลทกำลงถกสงระหวางผใชและเซรฟเวอรได ทำใหเกดการลกลอบเขาถงขอมลหรอปญหาดานความปลอดภยอนๆ

(security misconfiguration) เป็นการเกิดช่องโหว่ที่เกิดขึ้นเมื่อระบบหรือแอปพลิเคชันถูกกำหนดค่าหรือตั้งค่าไม่ถูกต้องโดยทำให้เปิดเผยต่อความเสี่ยงในด้านความมั่นคงปลอดภัยได้ ตัวอย่างเช่น Weak authentication and authorization settings - นี่สามารถรวมถึงการใช้รหัสผ่านเริ่มต้นหรือรหัสผ่านที่อ่อนแอที่สามารถทายได้ง่าย การไม่ใช้การตรวจสอบสำหรับหลายปัจจัยหรือการอนุญาตให้เข้าถึงข้อมูลที่เป็นความลับโดยไม่มีการอนุญาตเหมาะสม Improperly configured error messages - ข้อความผิดปกติที่ให้ข้อมูลมากเกินไปหรือเปิดเผยรายละเอียดของระบบสามารถถูกใช้โดยผู้โจมตีเพื่อเข้าถึงระบบโดยไม่ได้รับอนุญาต Misconfigured file permissions - สิทธิ์ไฟล์ที่ไม่ถูกต้องสามารถอนุญาตให้เข้าถึงไฟล์หรือข้อมูลที่เป็นความลับได้โดยไม่ได้รับอนุญาต Insecure default configurations - แอปพลิเคชันเว็บหลายๆ ตัวมาพร้อมกับการตั้งค่าเริ่มต้นที่ออกแบบมาสำหรับความสะดวกในการใช้งานมากกว่าความปลอดภัย การไม่เปลี่ยนการตั้งค่าเรมตน

`Vulnerable and outdated components หมายถึงส่วนประกอบซอฟต์แวร์ที่มีช่องโหว่ด้านความปลอดภัยที่รู้จักหรือไม่ได้รับการสนับสนุนจากผู้พัฒนาของตน ส่วนประกอบเหล่านี้สามารถเป็นได้ทั้งไลบรารี

เฟรมเวิร์ก หรือแพคเกจซอฟต์แวร์อื่นที่ใช้ในการพัฒนาระบบซอฟต์แวร์ขนาดใหญ่ขึ้น

การใช้ส่วนประกอบที่มีช่องโหว่และเลิกสนับสนุนอาจเป็นอันตรายต่อความปลอดภัยโดยมีผู้โจมตีสามารถใช้ช่องโหว่ที่รู้จักเพื่อเข้าถึงระบบหรือข้อมูลที่มีความสำคัญได้ นอกจากนี้

ส่วนประกอบที่ไม่ได้รับการสนับสนุนอาจไม่ได้รับแพทช์หรืออัพเดทเพื่อแก้ไขช่องโหว่ใหม่ ทำให้ระบบเปิดเผยต่อเวกเตอร์โจมตีใหม่

เช่น การที่ใช้version php เก่าที่มีช่องโหว่ที่ทำให้ผู้ปุกรุกสามารถโจมตีที่เข้าที่ช่องโหว่ โดยช่องโหว่เหล่านี้บางครั้งสามารถค้นหาผ่านโลกออนไลน์ได้ทั่วไป

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A07:2021-Identification and Authentication Failures

การล้มเหลวในการระบุตัวและการตรวจสอบตัวตนเกิดขึ้นเมื่อตัวตนของผู้ใช้ไม่ได้รับการยืนยันหรือตรวจสอบอย่างถูกต้อง ส่งผลให้เกิดการเข้าถึงข้อมูลหรือทรัพยากรที่เป็นความลับหรือสำคัญโดยไม่ได้รับอนุญาต

การล้มเหลวดังกล่าวสามารถเกิดขึ้นได้จากหลายสาเหตุ เช่น รหัสผ่านที่ไม่ปลอดภัยหรือง่ายต่อการเดา เปิดเผยรหัสผ่าน หรือนำรหัสผ่านเดิมมาใช้กับบัญชีอื่น ๆ การใช้เครือข่ายหรืออุปกรณ์ที่ไม่ปลอดภัยเพื่อเข้าถึงข้อมูลหรือทรัพยากร

ที่เป็นความลับหรือสำคัญ เช่น การเข้าถึงบริการธนาคารออนไลน์บนเครือข่าย Wi-Fi สาธารณะหรือการใช้ซอฟต์แวร์ที่ล้าสมัย ซึ่งอาจเปิดเผยผู้ใช้ต่อความเสี่ยงต่าง ๆ เช่น การติดเชื้อมัลแวร์ การฉ้อโกง (phishing) หรือการโจมตีกลางคัน

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Software and data integrity failuresกล่าวถึงเหตุการณ์ที่โปรแกรมซอฟต์แวร์หรือข้อมูลดิจิทัลเสียหาย ไม่ว่าจะเกิดจากสาเหตุใด ๆ เช่น ข้อผิดพลาดในการเขียนโค้ด บั๊กของระบบ

การโจมตีจากผู้ไม่หวังดี หรือความผิดพลาดของมนุษย์ ตัวอย่างเข่น ซอฟต์แวร์หรือข้อมูลมีการล้มเหลวด้านความสมบูรณ์ อาจทำให้เกิดการละเมิดข้อมูลส่วนบุคคล เช่น การแฮกหรือการโจมตีที่ทำให้ข้อมูลผู้ใช้ถูกเปิดเผย

ซึ่งอาจมีผลต่อผู้ใช้ บริษัท และชื่อเสียงของเว็บแอปพลิเคชันนั้น ๆ

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Security logging and monitoring failures

กล่าวถึงการตรวจสอบระบบและsecurity logging หากองค์กรไม่สามารถตรวจสอบระบบและเครือข่ายของพวกเขาได้อย่างถูกต้อง

องค์กรอาจพลาดเหตุการณ์ความปลอดภัยที่สำคัญและไม่สามารถตอบสนองต่อเหตุการณ์ความปลอดภัยได้อย่างทันเวลา

ซึ่งอาจเกิดจากสาเหตุเหล่านี

ขาดผู้เชี่ยวชาญที่คอยทำ monitoring และ ดูแลระบบภายใน ต่อมาอาจเกิดจากการขาดการตรวจสอบระบบ รวมถึงการconfig ที่ผิดพลาดส่งผลให้เกิด error ในระบบเป็นต้น

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

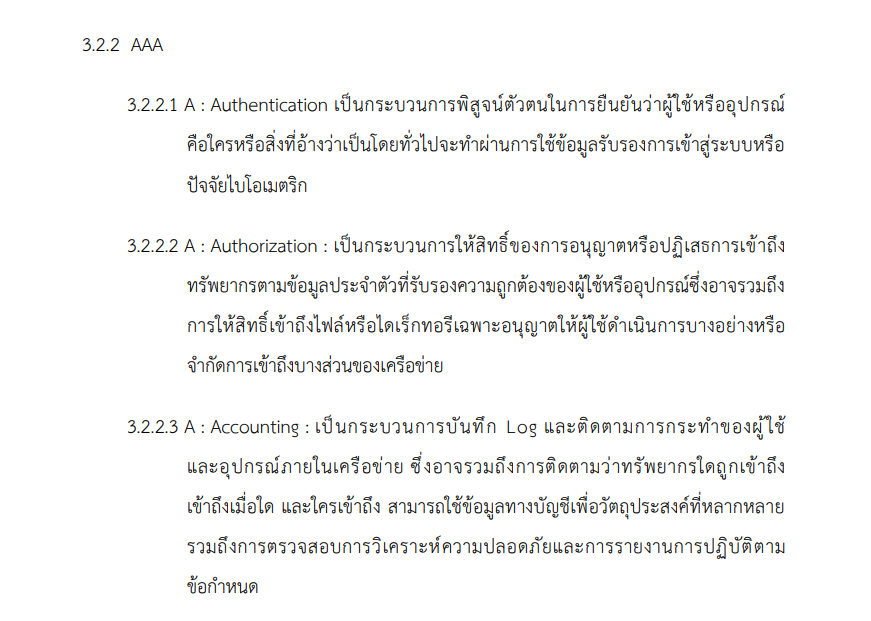
SSRF (Server-Side Request Forgery) เป็นช่องโหว่ที่อนุญาตให้ผู้โจมตีส่งคำขอที่เขียนมาเองจากแอปพลิเคชันด้านเซิร์ฟเวอร์ไปยังinternal system หรือ External system

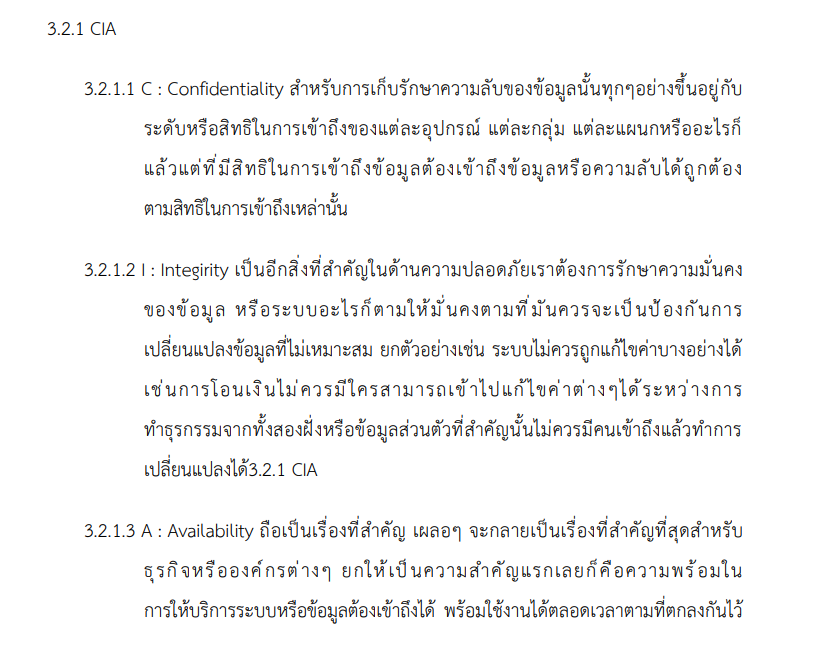
โดยการ bypassing access controlsและสร้างความเสี่ยงในการทำลายระบบ

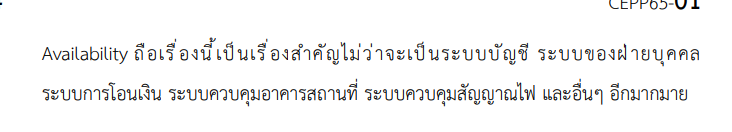
ตัวอย่างเช่น ตัวอย่างเช่น สมมติว่ามีเว็บแอปพลิเคชันที่อนุญาตให้ผู้ใช้ส่ง URL เพื่อดึงเนื้อหาของหน้าเว็บไซต์แล้วแสดงผลบนเว็บไซต์ แอปพลิเคชันนี้อาจใช้สคริปต์ด้านเซิร์ฟเวอร์เพื่อดึงเนื้อหาของ URL แล้วแสดงหน้าเว็บไซต์นั้นบนเบราว์เซอร์ของผู้ใช้งาน

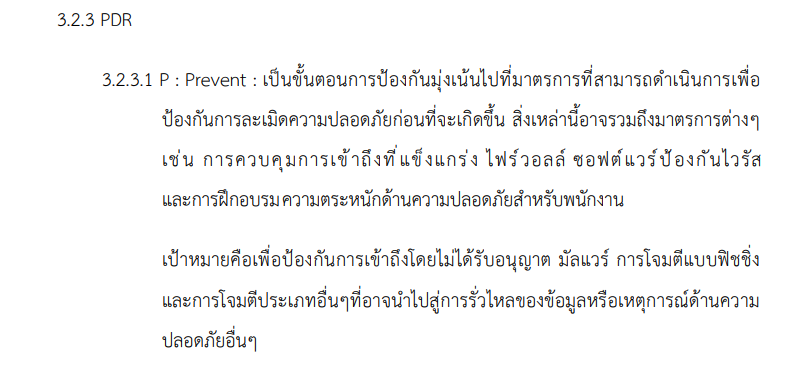
ผู้โจมตีสามารถใช้คุณสมบัตินี้เพื่อส่ง URL ที่ชี้ไปยังระบบภายใน เช่นฐานข้อมูลหรือเซิร์ฟเวอร์แอปพลิเคชัน ที่ไม่ควรเข้าถึงจากอินเตอร์เน็ต สคริปต์ด้านเซิร์ฟเวอร์จะดึงเนื้อหาของ URL ตามที่ร้องขอ แต่ผู้โจมตีสามารถแก้ไข URL เพื่อดำเนินการคำสั่งอย่างอ arbitrary หรือเข้าถึงข้อมูลที่สำคัญ

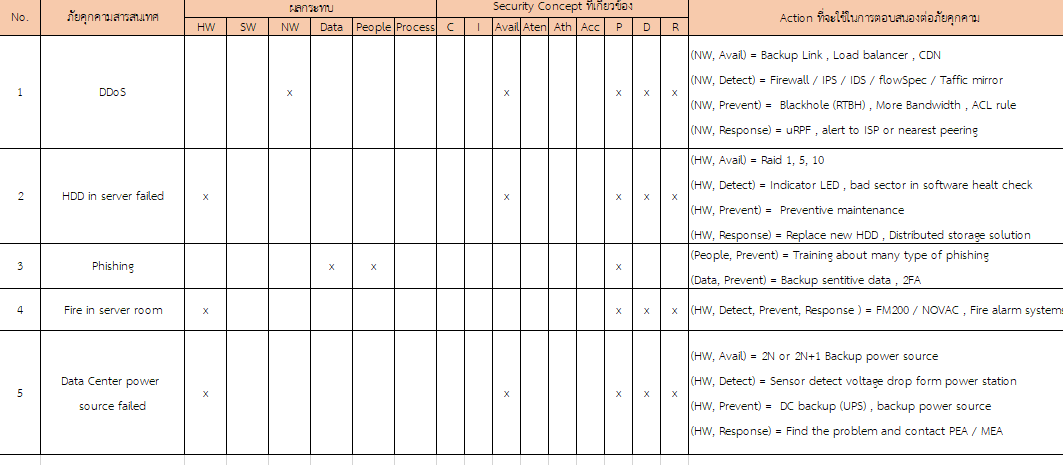
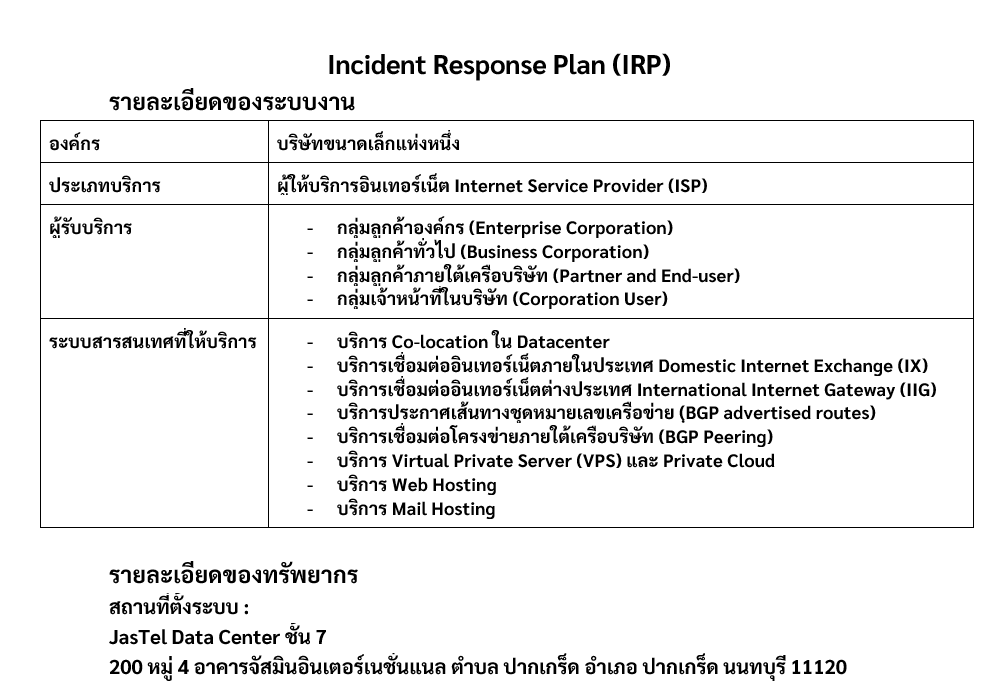
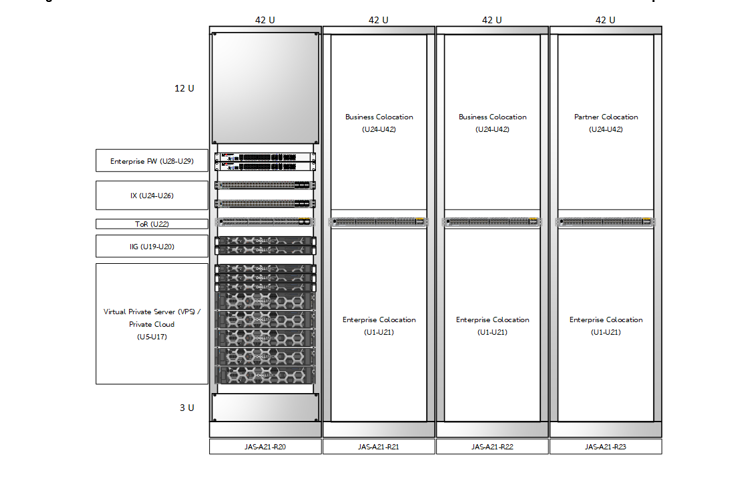
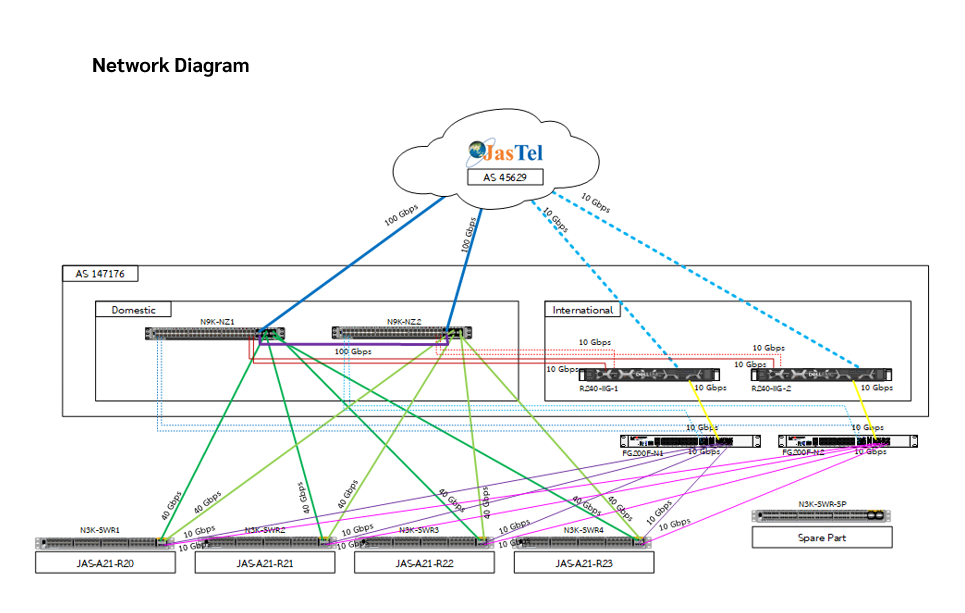
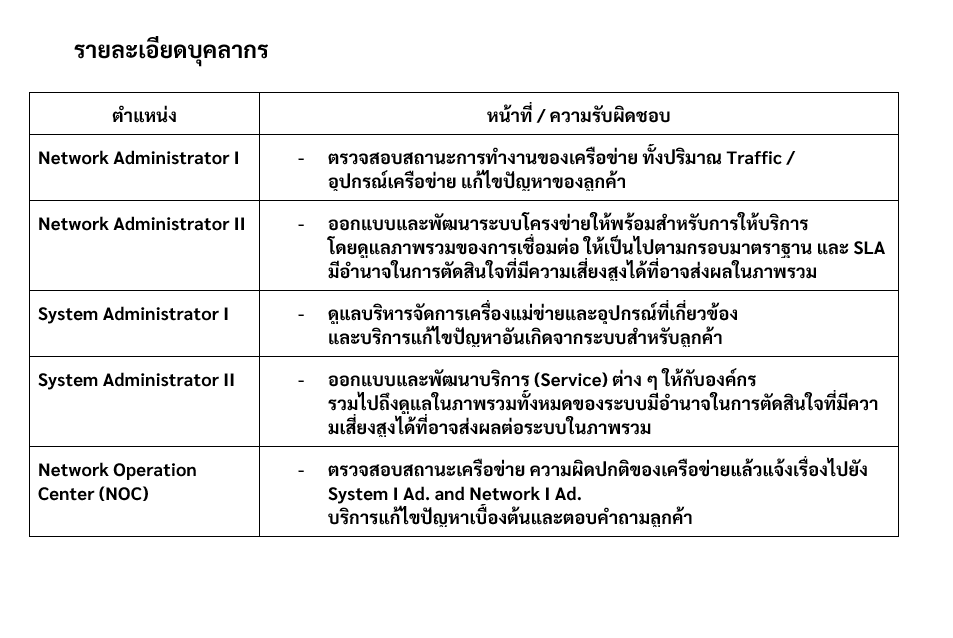
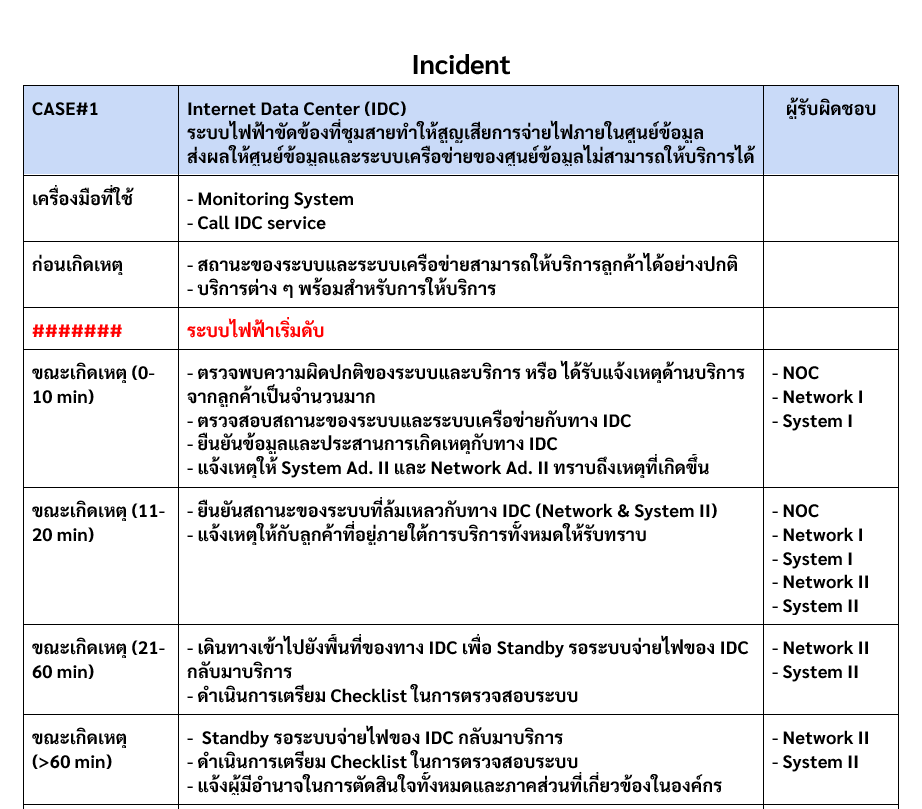
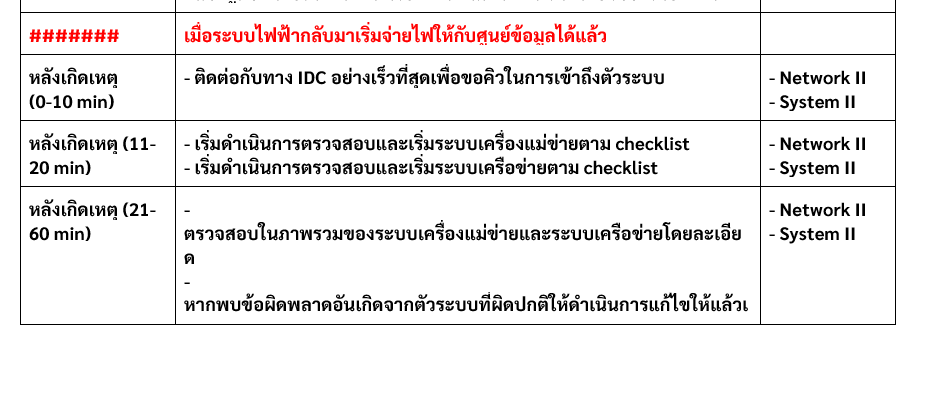
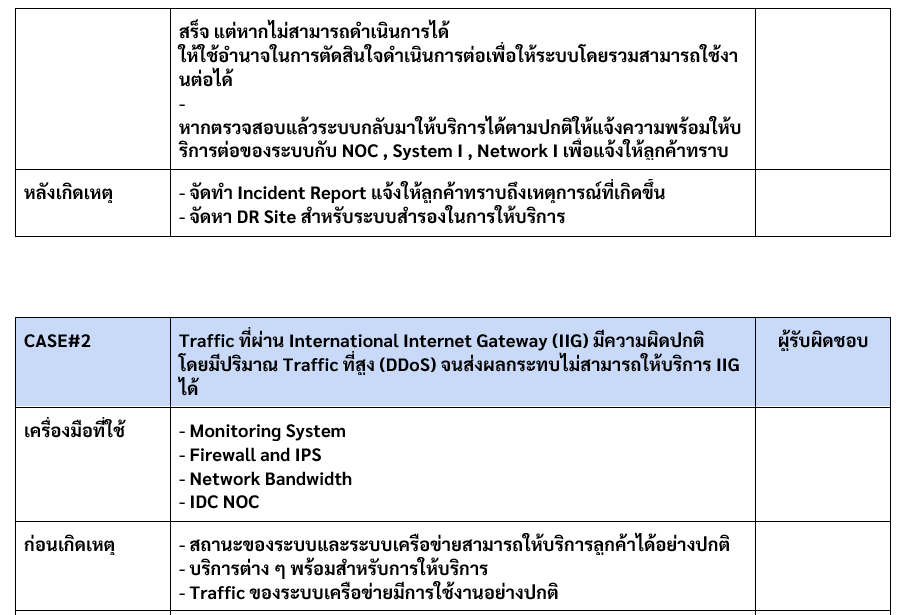
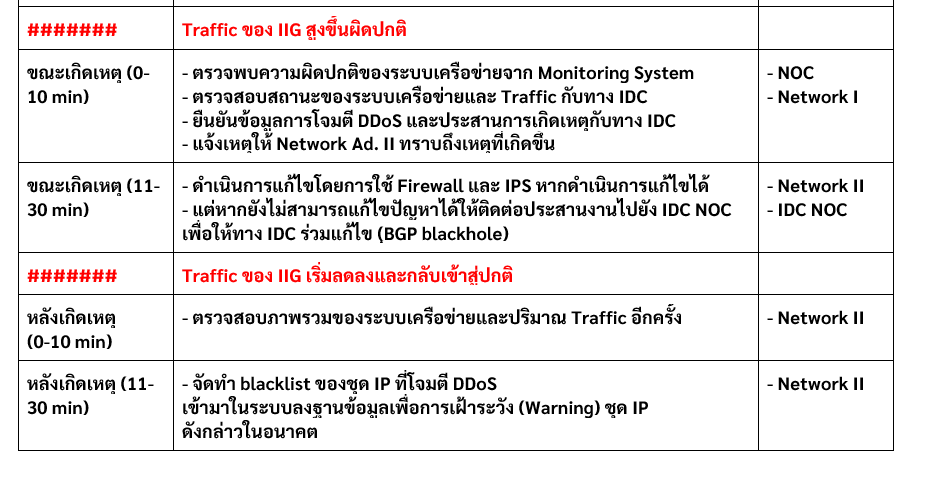
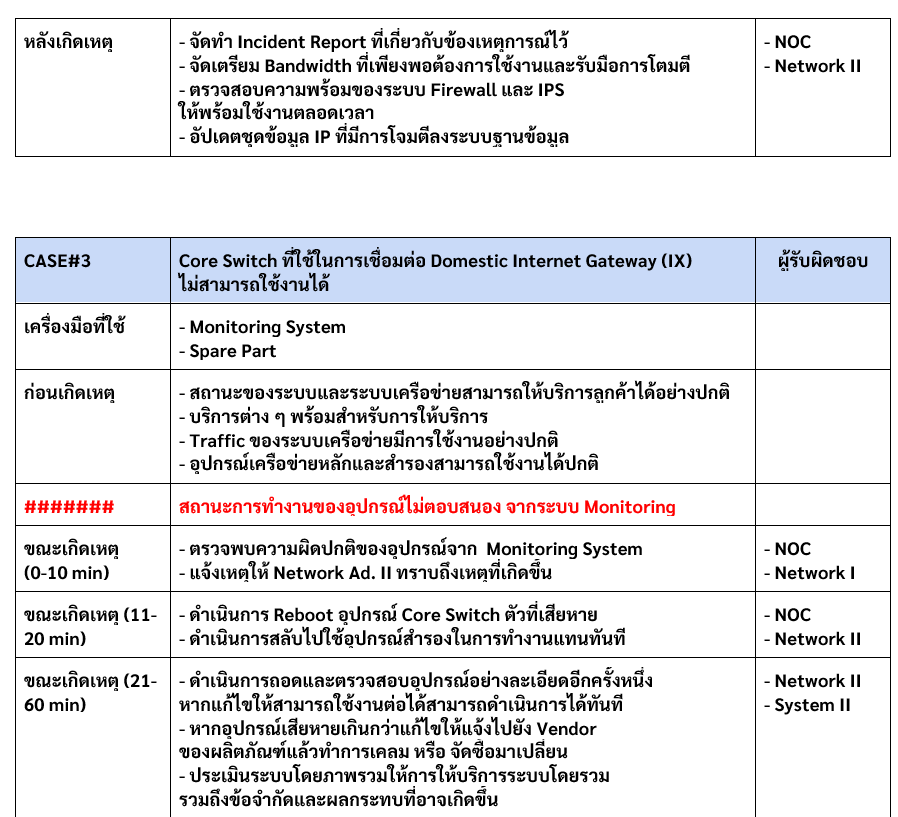
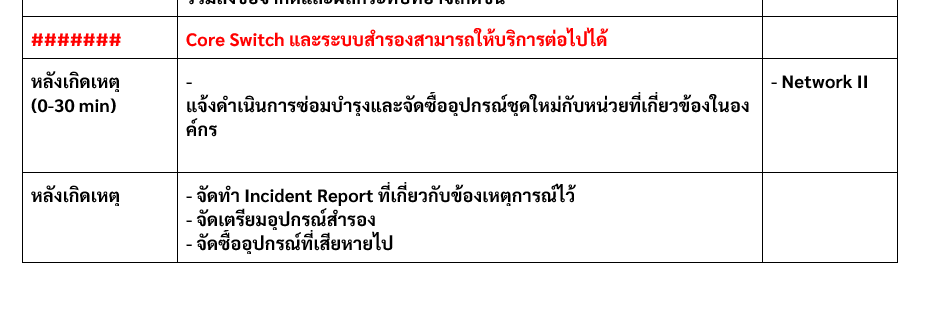
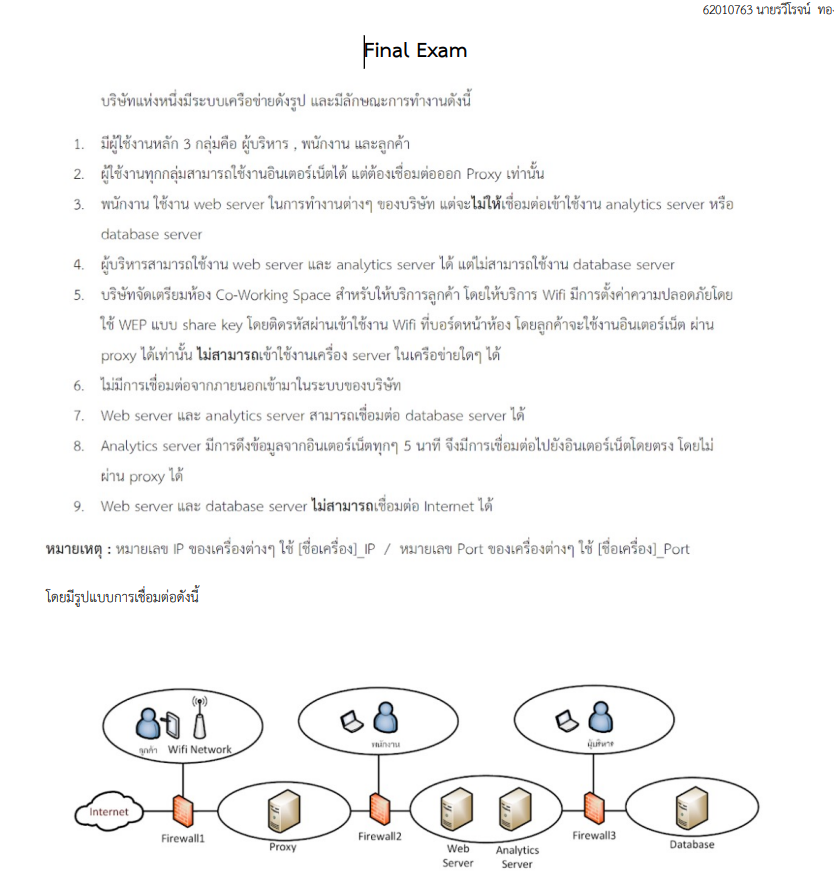
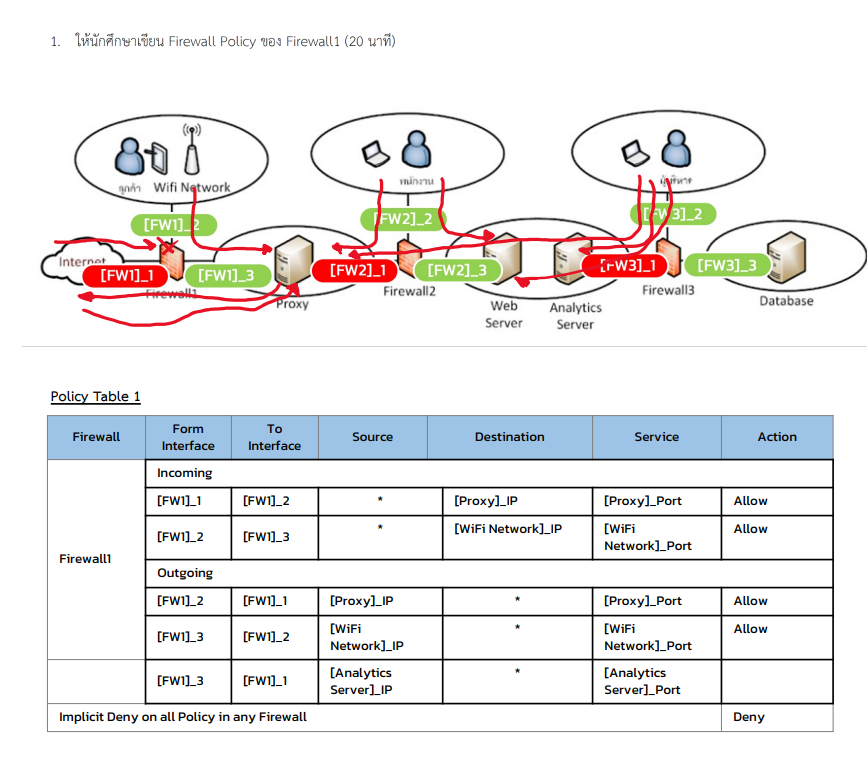
เช่น ผู้โจมตีอาจส่ง URL ที่รวม IP address หรือชื่อโฮสต์ของระบบภายในและหมายเลขพอร์ตที่ใช้สำหรับการดูแลรักษา ผู้โจมตีจะสามารถใช้ความเสี่ยงโหลดคำสั่งลงในระบบภายใน เข้าถึงข้อมูลที่สำคัญ หรือเริ่มการโจมตีต่อไปได้







             cxcds