娟娟 111.7.4

Domain 1. Security and Risk Management C、I、A+資安主管的基本功 GRC
Domain 2. Asset Security 盤點、分類、保護(RMF 懶人包，保護資訊資產)
Domain 3. Security Architecture and Engineering 時時(生命週期)都安全、處處(架構)都安全
Domain 4. Communication and Network Security 處處(架構)都安全
Domain 5. Identity and Access Management (IAM) I +3A 驗證身份(Authentication)、檢查授權(Authorization)及記錄行為(Accounting)
Domain 6. Security Assessment and Testing 查驗、訪談、測試；安全控制之有效性(目標)及符合性(需求)
Domain 7. Security Operations 安全維運、持續改善
Domain 8. Software Development Security 時時(生命週期)都安全、處處(架構)都安全

#	內容
資訊安全	透過安全管制措施，保護資訊資產免於受到危害，以達到 CIA 的目標，進而支持組織的業務，以創造價值，實現組織的使命與願景
資安三階目標	1. 一階目標：創造價值 2. 二階目標："支持組織的業務"，必須作到： ● 將安全融入組織的各個業務流程 ● 支持組織產品及服務的持續交付 3. 三階目標：Achieve CIA
治理的目標	創造價值
治理	公司最高經營階層的管理作為
管理	達成目標的一套有系統的方法；最常用的方法是 PDCA 問: 為啥要這麼做的意圖?
戰略	達成目標的高階方法(approach)或計畫(strategic plan) ■ 未來目標與現在程度的落差進行 Gap Analysis，就是 Road MAP ■ 需要 Resources 資源及遇到 Constraints 限制
戰略 3 元素	戰略目標由各項投資組合 Portfolio 組成(ROI) 投資組合 Portfolio 由計畫(Program)組成 計畫(Program)由 Project 組成
戰略管理	戰略形成 ： 如何想到 戰略執行 ： 如何執行 或是 戰略制定（戰略思維，外部和內部分析 SWOT，差距分析） 戰略實施/執行 戰略評估
業務企畫案、創新提案、候選方案、立案	業務企畫案 Business case(成本效益分析)--> 初期提出稱為創新提案 Initiative-> Alternative 候選方案-->立案 Charting
Scope creep	範疇淺變-不受控的範圍變化，係指專案進行過程中發生未經授權或未加以控制的範疇變更，即未走公司標準變更管理程序
Baseline	核定過的都是基準，需要調整要經過變更申請，如範圍、時間、成本、組態、資安基準，提供了整個組織中的每個系統都必須滿足的最低安全級別
Policy	書面的管理意圖；高階主管定義 Policy，讓執行者執行
Policy Framework(強制性,參考性)	強制性： 政策 Policies 最不能變動者 標準 Standards

	程序 Procedures、 SOP、 Step by step 參考性(非強制) : 指引 Guidelines 補充上述交代不清者
Security Function	組織處理資安相關事務的能力(資訊安全的定位和組織)、系統級別的“系統或系統元素提供的功能”
人事生命週期	Onboard，Promotion，Rotation，Projects，Passwords，Resign JD〔職責分離 (separation of duties)/工作職責(job responsibilities)/崗位輪換 (job rotation)〕、設定工作級別、篩選 candidate(Security clearance)、入職(雇傭協議/AUP/NDA/供裝(SPML))、訓練(意識/訓練/教育)、離職(面談/資產交回/反供裝/目送離開)
採購	合約生命週期 ■ 甲方招標文件 (RFI、RFP) ■ 說明會 ■ 投標 ■ 評選 ■ 議價 ■ 簽約 ■ 履約(合約):SLA/SLR, Security Requirement, Audit Right ■ 結案 丙方認證乙方能力 Capability：PCI DSS、ISO 27001、Star、CMMI 丙方以提供保證的第三方為主：軟體開發、取得採購/外包、服務交付能力 Assurance：提升信心、見證制度、稽核
CMMI	能力成熟度模型(不限於軟體) 3 能力：開發能力、服務能力、 採購能力 5 個能力成熟度： ①. Initial：土法煉鋼 ②. Managed：各自為政 (有專案經理，但各自為政/各 PM 的經驗不一) ③. Defined：統一方法 (公司立定統一一套辦法) ④. Quantitatively Managed：量化管理 ⑤. Optimizing 最佳化：持續改善
Business Continuity 7 個步驟	內外部環境分析：例如 SWOT 找出利害關係人及其需求 根據目標制定 Scope：根據組織結構與產品及服務兩個面向訂出來，以清單方式來呈現 BIA(關鍵流程及資源): 找出的**關鍵流程、關鍵資源**做風險管理(風險評鑑、風險處置) ①. MTD(Maximum tolerable downtime)：關鍵流程可以停多久(業務單位訂出) MTD = RTO + ?? ②. RTO(Recovery Time Objective)：恢復時間目標(業務單位給 IT 的目標) ③. RPO(Recovery Point Objective)：是指當故障發生時可以接受的最少資料損失，來評估多久進行一次備份，意指備份資料的頻率，備份頻率越多(間隔時間越短)可復原資料越完整； ④. SDO(Service Delivery Objective)：服務傳遞目的，在 RTO 時間內必須要達到的最低服務水準

	風險評鑑(識別、分析、評估)： ①. 識別：用來識別是否與目標有關，有關才進行分析 ②. 分析：分析的分是【分解】，析則是指【深入了解】;找出不確定性/影響，決定曝險值(Risk Explosion) ③. 評估：排序，根據風險分析結果，來決定是否進入處置階段。 風險處置 ATMA(事先、事中、事後) ①. 避免(Avoid)：放棄原本要做的事情 ②. 移轉(Transfer)：第三方，Ex.買資訊安全險或外包。 風險可以轉移，擔責任/法律責任不行 ③. 緩解(Mitigate)：(俗稱大事化小小事化無)，如 UAT 計畫能多跑幾輪與詳細列出聯外測試的情境，資訊系統備份資料的時間是否能再更頻繁，在與 stakeholders update 時的資訊是否充足且顧及到對方的反饋，這些都是可最小化風險的執行方向(寫計畫書) ④. 接受(Accept)：僅留在風險登錄表，考量成本效益分析 寫計畫書進行測試及演練： ①. Checklist ②. Read-through: 文件審查 ③. Walkthrough/Tabletop：穿行測試，角色扮演 ④. Simulation：模擬測試，直接演一演 ⑤. Parallel：平行測試，一組人到異地站點去，原系統不中斷 ⑥. Full interruption：全中斷測試
BC 標準(2 個)	ISO 22301 及 NISP
Risk	影響目標達成的不確定因素
風險三因素	1.目標 2.不確定因素 3.影響
風險管理標準	ISO 31000
風險管理(4 Steps)	1 大目標：確定目標、風險管理經營高層可以接受的程度 風險管理 2 大程序：風險評鑑／風險處置 風險評鑑 3 大步驟： ①. 識別(目標、風險登錄表 Risk Register) ②. 分析(算出，風險曝險值 Risk exposure，量化、質化) ③. 評估：判斷風險優先等級順序 風險處置 4 大絕招：規避 Avoid、移轉 Transfer、緩解 Mitigation、接受 Acceptance
風險管理目標	達到經營高層可以接受的程度
Risk Analysis Approach (2)	Qualitative 質化／定性 ：依賴主觀直覺、經驗和判斷，如：訪談法、德爾菲法和情景分析法 Quantitative Analysis 量化／定量： 年度損失預期值 (ALE) = 單一損失預期值 (SLE) × 年度發生比率 (ARO) 單一損失預期值 (SLE) = 資產價值 (Asset Value) × 暴露因素 (Exposure Factor)
Organization Resiliency Planning(計畫書)	Tier 1 組織層級 (Strategic planning) 策略規劃 ①. Crisis Communication Plan (CCP) 危機溝通計畫，緊急狀況 ②. Occupant Emergency Plan (OEP) 人員疏散計畫 Tier 2 Mission / 業務流程 ①. Business Continuity Plan (BCP)業務持續性計畫，人員(重要幹部、產線操作員出事)或疫情，但與技術無關

	②. Continuity of Operations Plan (COOP) 維運持續計畫，辦公區域損毀需在異地 30 天以上 ， 業務單位都要移轉，含繼位計畫 ③. Cyber Incident Response Plan (CIRP) 資安事故回應計畫 Tier 3 Information System ①. DRP Disaster Recovery Plan 事故回應計畫，所有事故應考慮(機房燒掉、異地機房、洪水來了)，強調資通安全事故 ②. ISCP Information System Contingency Plan 資訊系統應變計畫 ③. CIP Critical Infrastructure Protection 基礎設施(油、水、電、電信)
Recovery Site	①. Mirrored Site ，資料對拷 RTO：0-30s 恢復時間目標(Recovery Time Objective， RTO) RPO：zero ②. Hot Site 熱站點，與生產環境，只差最新一份資料 RTO：30s-30m RPO：zero ③. Warm Site 溫站點，解決採購設備問題(無資料) RTO：30m-72h RPO：>zero ④. Cold Site 冷站點，解決租賃地點問題 RTO：>72h RPO：>zero
Backup	Full Backup：most negligible overhead and the best efficiency in terms of restoration Differential Backup 差異備份，在完全備份之後有異動的資料就備份 Incremental Backup 增量備份，跟前一天比
資產保護	強調【資訊系統】 盤點：找產銷人發財各單位主管找出來，盤點資產的內容可能有生產配方、業務的目錄(產品照片-版權、商標唯一識別公司或產品)、員工資料(個資定義、處理、角色識別化)、研發資料(專利公開)；資產盤點完，首先要指定資產的擁有者(owner) 分類： ①. Business Value ②. Classification Scheme 不同資產有不同的分類表 ⊥ 美國軍方 (USCST)：極機密 Top Secret、秘密 Secret、機密 Confidential、未分類 Unclassified ⊥ 一般企業：機密 Confidential/ 私有 Private、敏感 Sensitive、公開 Public ⊥ People，背景調查後，核定機密等級(security clearance) 或職位 ⊥ NIST RMF 對資訊系統的分類：高/中/低衝擊 ⊥ 業務流程(processes)：關鍵/非關鍵流程或核心/非核心業務 保護：HIPPA、(ISC)2、RMF、ISO 27001 附件 A 144 控制項
Compliance	Organization Level 組織層面 ● Laws(主法機關) & Regulations(行政單位) ● Industry Standards 行業標準 (PCIDSS / ISMS) ● Contracts Requirements 合約要求 Individual Level 個人級別

	• Organizational Policies 組織政策 • Due Diligence / Due Care 盡職調查/應有關注 • Ethics 道德
Assets	硬體、OS、軟體、網路、機房、人員、業務流程
FISMA	Federal Information Security Management Act 資訊安全根本大法
SOX	Sarbanes-Oxley Act 監督內線交易，監督受證券法約束之上市公司的審核及相關事務，以保護投資者利益
GLBA	Gramm-Leach-Bliley Act 集團內的客戶個資做規範。允許商務和投資銀行、證券公司和保險公司進行整合，並解決保護使用者隱私權的問題。
HIPAA	Health Insurance Portability and Accountability Act 促進醫療健康產業善加利用新科技，並為醫療資訊的安全和隱私建立屏障
GDPR	歐盟一般資料保護規範： 個資的認定：直接或間接若為當事人唯一識別 GDPR 角色定義 Controller 決定蒐集目的與處理方式、Processor 代表控制者，根據其目的與處理方式來處理個資、Data Subject 個資的處理原則： ①. 蒐集時告知目的 ②. 取得同意 ③. 蒐集最小化（用不到不蒐集） ④. 善盡保護原則(De-identification) ⑤. (怎麼用) 兼顧資料品質:開放客戶修改、刪除資料 ⑥. 出事要告知 / 出事要負責（賠錢/法律責任無法卸責），爆發個資外洩的資安事件時，必須要在 72 小時內，即刻通報給資料保護主管機關（Data Protection Authority）
版權	版權法保護原創作品的創作者，防止創作者的作品遭未經授權的複製。 Copyright protects only the **expression of ideas** and not ideas themselves. ①. 創作者，版權歸作品的創作者所有：被保護時間，第一作者去世後的 70 年 ②. 受雇用而創作的作品：第一次發表日後的 95 年，創建之日 120 年取短 ③. **數字千年版版權法 DMCA**：防止複製數字介質（數位媒體）、若罪犯使用 ISP 線路嘗試違反版權活動時，ISP 要負責
商標 Trademark	主要目的用來識別主要企業或產品 Ex. M 麥當勞 有登記 -> ® 、未登記 -> TM 聯邦商標法（Lanham Act）:《**蘭漢姆法案**》是美國主要的聯邦商標法。該法禁止許多活動，包括商標侵權，商標淡化和虛假廣告
營業秘密 Trade Secret	企業中善盡保護不讓人知道的資訊，必須確保存取相關資訊的人簽有 NDA，如：可口可樂配方、程式碼（秘密性、經濟價值性、及合理保密措施） 美國**經濟間諜法**(The Economic Espionage Act of 1996)：以政府之公權力來保障私人企業之智慧財產權
專利	發明或創作，為維護其權益，向智慧局提出申請，經過審查認為符合專利法之規定，而授與專利權，如:演算法比較可能登記專利 **創新**意味著專利

Data Role （企業）	Owner（擁有者)當責（動口） Steward（管理資料品質） Custodian（保管人）如 IT 人員、設定防火牆保護他、備份（動手）
Data Role （個資）	Controller 決定蒐集目的與處理方式 Processor 代表控制者，根據其目的與處理方式來處理個資 Data Subject / Principal 當事人
個資 De-identification 去識別化	Anonymization 匿名化的資料不是個資，匿名化指直接或間接都無法回溯 Pseudonymization 擬匿名化：可回朔的匿名化(若有用對照表或編號還是可以還原） *注意→對於使用者與測試人員是無法回朔的
資料三態	從 Data Stage 狀態來看時間資料三態 ①. 儲存 at rest（Storage)：對稱式加密、Bcrypt 對 Linux 密碼加密 ②. 使用 in use（載入記憶體）：顯示在螢幕上可用浮水印/防偷窺/截圖警告、教育訓練、DLP ③. 傳輸 in transit（封包擷取)：HTTPS 傳輸使用 TLS 作為底層加密協議、VPN(TLS、L2TP/IPsec)
Data 生命週期	①. Create 建立(企業內部資料) / 蒐集(個資) ②. Use ③. Share ④. Archive 封存 ⑤. Destroy
Sanitization 資料清洗	①. Clear 擦除，人員/實驗室可恢復，一般清洗(Clear) 透過指令 或 UI 操作: Erase，rewriting，rest，Remote wipe ②. Purge 清除，廠商提供的工具，需送實驗室才能救回 ■ Cryptographic-Erase 加密擦除(加密後並把金鑰刪除，適用雲端，或是 SSD，USB Flash base) ■ Block Erase ■ Degaussnig：消磁(適用磁性硬碟、磁帶) ■ OverWrite 是 purge 的一種技術手段，但要看寫入的次數與 Pattern ■ ATA sanitization I/O commands: ③. Destroy 銷毀，數據無法恢復和媒體物理性破壞，如碎化，美國國家安全局要求對 SSD 進行物理銷毀。這個過程稱為 disintegration 分解，通過 fragments 切碎過程產生非常小的碎片
SOC	①. SOC 1(又稱 SSAE16，前身為 SAS 70)，與財務報導相關的內部控制活動 ②. SOC 2 Report 則是針對企業組織之 Security、Availability、Processing Integrity、Confidentiality、及/或 Privacy 等五大領域（可自行選擇組合）的控制評估 ③. SOC 3：SOC2-Type2 精簡版，較少技術細節的內容，對外公開的服務報告 ✓ SOC 3 沒有類型 ✓ Type 1，文件審查(自說自話)，Type 1 is one or more times of examination (snapshot) of the suitability of design of controls. ✓ Type 2，書審+實地查核一段時間(6 個月)，其 Type 2 更可靠，因為包

	含對控制措施的獨立測試，Type 2 the examination of operation effectiveness of controls over a period of time.
SAMM	Software Assurance Maturity Model 軟體保證成熟度模型 成熟度分級如下： 0: 起步點，沒有達到任何目標 1: 有初步認識，達到部分目標 2: 加強功能達成的有效性 3: 對功能有深入的了解和掌握，能大規模地推行功能的要求 *設置安全冠軍：加強各個團隊與資訊安全之間的關係
Common Criteria	資訊技術安全評估共同準則（Common Criteria for IT Security Evaluation， ISO/IEC 15408），是針對實現資/通訊產品所使用資訊技術的安全性所進行的安全技術認證。 角色： ①. Target of Evaluation，TOE，廠商研發送驗的 IT 產品 ②. Security Target，ST，檢驗標準（標準廠商自己訂，第三方驗證） ③. Protection Profile，PP，範本，供 ST 參考 ④. Common Criteria Testing Lab，CCTL，CC 檢驗實驗室 - 評估保障等級（Evaluation Assurance Level, EAL）：以數值方式，每一個 EAL 會對應一組預先定義好的安全保障需求（SAR），這些安全保障需求涵蓋產品開發的全部過程，有一定的嚴謹性。數字越大代表越嚴格，但不是 EAL7 就比較安全 - 廠商連繫 CCTL，針對 TOE 之檢驗標準及範本進行驗證 Evaluation Assurance Level P.306 ● EAL7，Formally 正式(學術水準/數學理論)：產品基於有限狀態機設計 ● EAL6，Semi-formally 半正式 ● EAL5，半 Semi-formally 一半的半正式 ● EAL4，Methodically 有設計：產品基於高凝聚力，低耦合架構開發 ● EAL3，Methodically 有條理：在產品工程團隊的支持下對產品進行測試和檢查 ● EAL2，Structurally(白箱)有結構 ● EAL1，Functionally(黑箱)有功能：產品有效運行，如產品手冊/說明書中所述，不需要供應商工程團隊與 CC 實驗室合作
PCI DSS	The Payment Card Industry Data Security Standard 是國際支付卡品牌基於為支付卡產業保障持卡人資料安全所共同建置的全球統一規範。所有從事持卡人資料之保管、處理、傳輸的機構，均須關注其組織是否符合 PCI DSS。 PCI DSS requires that **rescan**（應該是 Code review) the application at least annually and after any change in the application
CSA/ STAR	STAR 雲端認證 = ISO27001 + 雲端控制矩陣 +成熟度評估
Data 保護 (ISC)2	**事前** ①. 指示類 Directive（行政管理類）：管理意圖表現 ②. 嚇阻類 Deterrent（告知你後果，不要這樣做）：打消動機 ③. 預防措施 Preventive（Ex. 門禁管制）：提高門檻 **事中**

	①. 偵測措施 Detective（一直測密碼，看是否有非授權情況） ②. 矯正措施 Corrective（保全來看）：出現問題或產生缺失後的處理或補救，目的在於防止不符合再發生 事後 ①. 復原措施 Recovery（已被破壞） ②. 其他-補償措施 Compensating（現行的不好用）： 針對某些環節的不足或缺陷而採取的控制措施
Data 保護 HIPAA 法案(3)	①. 行政管理類 Administrative：安全策略／程序，人員招募、背景調查，意識訓練，分類、標記 ②. 邏輯/技術類 Logical：身分驗證（密碼、SmartCard），受限接口 Restricted interface，ACL，防火牆，IDS ③. 實體類 Physical：fence 圍欄，alarms 警報器，Cameras 攝影機
Data 保護 NIST RMF*：7 項 PCSIAAM	prepare， **categorize system**， select controls， implement controls， assess controls， **authorize system**， monitor controls *是框架(懶人包) 僅是基準而已，所以不含風險評鑑
Scoping & Tailoring	範圍界定(Scoping)：排除不適用 裁縫(Tailoring)：訂制(量身訂作)
A&A(C&A)	RMF Access Controls and Authorize System (A&A) process C&A (Certification and Accreditation) is replaced by A&A (Access Controls and Authorize System) in RMF and V&V (Verification and Validation) in ISO 15288
身份驗證三步驟	Identification (1.出示身分 Credential = Identity 帳號 +Authenticator 密碼(認證))(告訴人家我是誰) ● 使用者輸入帳號/密碼 ● 按下送出 Validation (2.檢查授權)：收到帳號/密碼到帳號資料資料庫 Directory 比對 Notification(3.通知結果) ：Server Side 驗證後回覆 Access Token 給 Client (授權的基礎)，後續 Client 出示 Token 去存取資源
SAML/OIDC	Token 內有 斷言 "Assertion" (SAML) 或 宣言 Claim (OIDC，OpenID Connect)，是 Authentication server 對通過身份驗證的 subject 的肯定描述，技術上的資料結構通常是一個 Key-Value Pair，也就是一個實體的屬性加上這個屬性的值（屬性 = 值）。 SAML (Security Assertion Markup Language)XML 基礎 <ID>Jack</ID> OIDC (OpenID Connect) - JSON 基礎 { ID : 'Jack'}
SSO	使用者登入一次（不是指一個帳號），即可跨系統存取資源（保留自己的帳號），一次登入可以對應多個系統帳號（每個系統都有自己的帳號） ①. 整合型（Integrated)：微軟 AD，一個使用者只有一個帳號 ②. 聯盟式（Federated Identity)：使用者在每個系統都有自己的帳號，透過聯盟的關係，使用盟主的帳號即可跨系統登入；On-premise/IDaaS ③. 老系統，可寫程式幫忙登入（讓使用者感覺只登入一次），腳本/logon script 無法 SSO 的話，可簡化登入程序，如記憶密碼
MFA	Something you know：User Account and password

	Something you have：ID Cards and OTP Tokens Something you have：Biometric Template
生物辨識	Physiological - Face - Hand：Fingerprints 指紋、Finger-vein 靜脈、Palm 掌紋 - Eyes：Iris 虹膜、Retina 視網膜（容易遭疾病（糖尿病）影響、精確度高(非外顯)） ● Behavioral：Voice、Signature 簽名動態、 Keystroke dynamic 鍵盤動態 ● Biological：DNA、Blood Glucose 血糖
Authorization 協議	**Extranet - Authentication protocols** XACML（SAML） 基於風險和基於屬性的存取控制是授權機制 ➢ eXtensible Access Control Markup Language - XML 基礎 ➢ 應用場域 ： 如 供應鏈 嚴謹 ➢ PEP、PDP 概念設計 ➢ 策略決策點（PDP）是 XACML 中提到的支持基於屬性的訪問控制的核心組件之一 **Internet - Authentication protocols** OAuth2（OIDC：OpenID Connect）：一個開發標準（Open Standard）用來處理有關「授權」(Authorization）相關的行為， JSON 基礎 ➢ 使用 OIDC 的 OAuth 2.0 客戶端(Clients)也稱為依賴方(Relying Parties, RPs) ➢ 實現 OIDC 的 OAuth 2.0 身份驗證服務器也稱為 OpenID 提供者 (OpenID Providers, OPs)
DAC (Discretionary Access Control)	①. 隨意型存取控制，隨 Owner（File Owner/Data Owner）的意思，做授權決策，可稱 Identity-based 身份型(在系統上設權限、用帳號設權限)，由 Custadeo 實施 ②. need-to-know（工作職務有需要才需要知道）、least privileges(權限給不多也不少) ③. Graham-Denning Model ④. 存取控制矩陣（Access Control Matrix，ACM）lists objects, subjects, and their privileges ➢ 資源的存取表（Object，被動方，左右欄位），Access Control List(ACL) ➢ 人的權限存取表（Subject，主動方，上下欄位），從人的角度（能力表）Capability Table ➢ Take-Grant Model，能力表操作（ACM 新增移除）理論
Role-based Access Control	以角色（具有權限的群組 Group→Windows）為基礎。角色有帶權限，群組無權限概念，綁定職務，可消除特權潛變 Privilege Creep，強制執行 Least Privilege
Attribute-base Access Control	ABAC，屬性為基礎的存取控制，現代權限控管的主流，綜合 Subject 主體的屬性（ID、性別）、Object 客體的屬性（IP、系統）和環境的屬性（時間），訂定複雜的授權規則。比較符合實際業務需求（特殊促銷案），協定：XACML - 分為條件式(criteria-based)及分數制(score-based)

	⊥ 零信任 ⊥ 802.1X ⊥ typically implemented in software defined networks (SDNs) ⊥ 在 XACML 中，負責檢查授權的系統(如 web server)稱為 PEP，PEP 可向 PDP(專屬的授權伺服器)查詢主體是否有被授權。Custodian 可在 PAP(Policy Administration Point)上作權限的設定，因為 ABAC 是以屬性為基礎，實施條件式或分數制的授權，所以由 PIP(Policy Information Point) 負責收集其它屬性資料 ⊥ implemented in software defined networks (SDNs)
Risk-based Access Control	風險為基礎的存取控制，ABAC 的延伸，依風險（屬性）分數。(本人 1 分、上班時間 2 分、公司電腦 2 分，存取系統需 10 分以下)。
Rule-based Access Control	以規則(如 ACL 中的規則)為基礎的存取控制，if-else，等同防火牆、統一控管、集中控管
Mandatory Access Control	強制型存取控制。**系統根據標籤**進行強制控制。人員（Subject）經過背景調查 Background Check and 安全檢查 Security clearance，資料（Object）進行分類，進行標籤。又稱 Lattice-based Access Control。 軍方人員等級與可存取專案-->BBCC 三種基於格的訪問控制模型（Bell-LaPadula、Biba 和 Chinese Wall）
Security Model(BBCC) BLP Bell-Lapadula Model	BLP Bell-Lapadula Model 為軍方設計 目的：確保**系統機密性** 解決**資料洩密**的問題 **橘皮書**的核心理論 **禁南下政策** (No Read UP ， No Write Down) 管制資料流動 一個**有限狀態機**，其中所有狀態都是**安全狀態** Simple 簡單屬性（管制讀取）：Simple security property Star 星號屬性（管制寫入）：* security property 在存在受信任主體的情況下，Bell-LaPadula 模型可能會產生從高機密文檔到低機密文檔的信息流動，受信任主體不受*屬性的限制
Biba Model	目的：確保**完整性** 禁北上政策 (No write up，no read down) 低層級不能寫到高層級 Simple 簡單屬性（管制讀取）：Simple Integrity property Star 星號屬性（管制寫入）：* Integrity property *補充 Bell-Lapadula and Biba models are NOT mutually exclusive in a trusted computer system. For example, The **Lipner model** combines elements of Bell-LaPadula and Biba.
Chinese Wall(Brewer Nash Model)	目的：確保系統機密性 設定**利益衝突**群組，根據**歷史**讀取紀錄，**動態**阻絕(動態權限)
Clark-Wilson Model	目的：確保完整性 以交易為基礎目的：資料庫/最小工作單位/全部成功或失敗 職責分離 SOD：敏感的工作分為二個或多個員工執行 ACID：是指資料庫管理系統（DBMS）在寫入或更新資料的過程中，為保證事務（transaction）是正確可靠的，所必須具備的四個特性：原子性（Atomicity，或稱不可分割性）、一致性（Consistency）、隔離性

	（Isolation，又稱獨立性）、持久性（Durability） 訪問三元素：subject（User），程序（TP），object（CDI） 這種模型使用了下列元素： ①. 用戶活動個體。 ②. 轉換過程(Transformation Procedure，TP）可編程的抽象操作，如讀、寫和更改。 ③. 約束數據項(Constrained Data Item，CDI）只能由 TP 操縱。 ④. 非約束數據項(Unconstrained Data Item，UDI）用戶可以通過簡單的讀寫操作進行操縱。 ⑤. 完整性驗證過程(Integrity Verification Procedure，IVP）檢查 CDI 與外部現實的一致性。(完整性)
Zero Trust	存取控制 2.0，以資料為中心，進行更細緻、動態及透明的存取控制： ● Security Principles：Need-to-Know、Least Privilege ● No inherent Trust ➢ Give up "Trust, but verify" ➢ Perimeterless ➢ Micro segmentation（微分段）：減低 Lateral movement attack 網絡橫向移動(在內網到處亂跑) ➢ Software-Defined Network（SDN） ● Continuous Verification ➢ Verify and never trust ➢ Network Access Control（802.1X） ➢ Mutual-Authentication ● Data-Centric 以數據為中心 ● Fine-grained 細粒狀（做的很細緻） ➢ Criteria-based vs Score-based（NIST） ✧ Criteria-based 就是企業定義好每個來源(帳號/設備/服務等)的要 Criteria-based 求被事先定義好它能執行的動作(例如讀取或寫入) ✧ Score-based 要求來源的各項資訊打分數外加企業可能已經定義好對這一項來源的分數，只要分數高於設定好的值，來源要求就會被通過。但如果分數沒有達到的話，原來定義好的允許動作還會被降低 ➢ Attribute-based vs Risk-based（ISC2） ➢ XACML 授權機制是基於風險和基於屬性的存取控制 ● Dynamic ➢ Port knocking（傳輸層）：讓所有的連接埠在一開始都不要開啓，然後以連接埠的組合設定一個暗號，只有知道暗號的人才能讓連接埠開啟，然後連線 ➢ Single Packet Authorization（SPA）伺服器完全防禦機制，其中只需要一次“敲門”，傳送加密過的 Single Packet ➢ Subject, Object, and Environment（ABAC 主體、客體、環境） ● Transparent（Visibility） ➢ Logging （Accounting）and Recording ➢ Monitoring and Inspection
Remote Network ：	①. Point-to-Point Protocol（PPP）是建立二個相鄰節點間之連結的協

PPP - Authentication protocols	定（點對點的網路協定），屬 Data Link Layer，不是身份驗證協議本身，但它制定了 PAP 及 CHAP 二個身份驗證協議來決定是否接受用戶端的連線。 ②. PPP 協定中的一個重要特性是，可以與各種不同的網路協定互相整合 - PAP：密碼驗證協定，是明碼傳送使用者的帳號及密碼，傳輸過程沒有加密，改良版是 SPAP - CHAP：挑戰與回應協議（MD5） ✧ Repeatedly：過程反覆驗證 ✧ 改良版：MS-CHAP ✧ CHAP 在身份驗證的過程沒有直接送出密碼，而是由 Server 端送出挑戰，由 Client 端結合帳號及密碼後以 MD5 取雜湊值；Server 收到後再重新計算雜湊值並比對結果，以達到驗證身份的目的。這個挑戰與回應的過程不只是 Client 剛撥接時會驗證一次，再傳輸的過程會不斷重覆的進行，如五分鐘一次 ③. EAP 擴展認證協議(Framework)：屬框架，非身份驗證協定，不是直接拿來作身份驗證，而是讓廠商開發身份驗證協議，often used for wireless networks ■ EAP-TLS ➢ 基於憑證驗證 ➢ Server 及 Client 都需要安裝憑證 ➢ Mutual authentication ➢ 憑證會到期，管理負擔重 ■ EAP-TTLS ➢ 精簡版 EAP-TLS ➢ 僅 Server 安裝憑證 ■ PEAP ➢ 微軟版 ➢ 改良 EAP-TLS ➢ 僅 Server 安裝憑證 ➢ it can provide a TLS tunnel that encapsulates EAP methods, protecting the entire session ■ EAP-MS-CHAP
Remote Network：VPN - Authentication protocols	①. VPN 就是透過 tunneling 技術，把公眾/共用網路當作(模擬成)網路線來連接公司的電腦及網路，由於使用公眾/共用網路，因此必須搭配安全服務(如身份驗證、金鑰交換、加密、完整性控制)才能安全地在 tunnel 上傳輸資料 ②. VPN 二大主題 = 建立私有通道(tunneling) + 提供安全服務(security services) ③. Tunnel：A 到 B，有一虛擬（Virtual ）連線 ④. VPN 主要的應用場合： ➢ Site-to-Site (Gateway-to-Gateway)：高雄分公司連接台北總公司 ➢ Remote Access/Dial-up (Host-to-Gateway)：在家上班，使用 VPN 撥接到公司 ➢ PPP 可用的 Protocol 都可以用

	⑤. 建立通道協議 Tunnel ➢ T2F（Cisco） ➢ PPTP（微軟）點對點隧道協定（Point to Point Tunneling Protocol）是由 MPPE 加密 ➢ L2TP（L2 Tunnel Protocol）(RFC) PPP 驗證協定，最常搭配 IPsec
IPsec	①. AH（Authentication Header）真實性 ②. ESP（Establishment of Encapsulating Security Payload），真實性+機密性，是 IPsec 體系結構中的一種主要協定，其主要設計來在 IPv4 和 IPv6 中提供安全服務 ➢ Transport mode：通常是直接建立在兩台主機上，因為不需要再多加一個 IP header ，整體來說較省頻寬，在這個模式下，兩邊的主機都要安裝 IPSec 的 protocal，而且不能隱藏主機的 IP 位置 Tunnel mode：針對 Firewall 或是 Gateway proxy，一般來說我們會用這個模式，因為他們不是原本的發送收端
Network Access Control-802.1X（身份驗證協議）	Local Area Network-Authentication protocols： ①. 區域網路(LAN)的網路存取控制(NAC, Network Access Control)，也就是在連接 AP 基地台或 Switch Hub 前要通過身份驗證才能完成連網 ②. 連線加密協定身份驗證協定：使用 PPP 協定 ③. 802.1X（身份驗證協議）：Client to Switch/AP 的身份驗證(如何在 LAN 裡面用存取控制就是 802.1X) ④. 802.1X 不要跟 VPN 搞混，VPN 是屬於 WAN 的機制，身份驗證不適用 802.1X（LAN 的驗證） ⑤. 802.1X 的身份驗證使用 EAP 系列的身份驗證協議（如 EAP-ELS， EAP-TTLS 及 PEAP 等），因此又稱為 EAP over LAN (EAPoL)。要特別留意 EAP-TLS 的 Mutual Authentication 對於 PKI 及憑證的需求；所有使用 EAP-TLS 的 Supplicants 及 Authenticators 都要安裝憑證，因此會造成系統管理上的負擔 ⑥. 使用 802.1X 實施網路存取控制中，讓請求者(supplicant)向身份驗證者(authenticator)進行身份驗證且具有最少的系統管理負擔(overhead)的是 PEAP 協議，Authenticator 跟 Authentication Server 之間多半是以 RADIUS 溝通；角色有：Supplicant 用戶端、Authenticator、Authentication Server 情境: ➢ Ethernet(based on the IEEE 802.3 standard)身份驗證（Switch，RADIUS Client） ➢ Wireless 身份驗證（AP，RADIUS Client）
RADIUS	1. RADIUS：AP 與 AD 認證的協定、AD 預設停用 2. *Network Access Servers and the Authentication Server per RADIUS 3. *RADIUS 是一種用於提供 authentication 和 authorization 的 AAA 協議；它通常用於 modems、無線網絡和網絡設備 4. *By default, RADIUS uses UDP and only encrypts passwords. ①. *RADIUS supports TCP and TLS, but this is not a default setting.

	②. *RADIUS supports TLS over TCP. 5. *RADIUS does not have a supported TLS mode over UDP
Intranet-Authentication protocols	①. NTLM (NT4) ②. Kerberos：Computer-network authentication protocol (Win7) P.217 ✧ Client ✧ KDC (Key Distribution Center) = 密鑰分發中心 ■ AS (Authentication Server) 認證伺服器、驗證身份 ■ TGS (Ticket Granting Server) 票據授權伺服器 ✧ Application Server ■ 認票 TGT ■ TGT (Ticket Granting Ticket) = 票據授權票據，票據的票據 ✧ 運作方式 ■ Client 透過送出 User ID(Cleartext)與 AS 進行認證後取得 TGT、Client/TGS session key ■ Client 提供 TGT 給 TGS 驗票，取得 Application Server Ticket ■ Client 提供 Ticket 給 Application Server 確認後，提供服務 ■ *用戶端使用明文傳送 identity (但不包含 password) 給 AS，只要 AS 在帳號資料庫能查到這一個帳號(identity)，基本上就算通過身分驗證，AS 就會發出一個回應訊息叫做 KRB_AS_REP, 把 TGT(有加密/對稱)傳給用戶端。被加密的是 TGT，不是用戶端的密碼，因為密碼根本沒有被傳送，就是用戶端在提出身分驗證請求的時候，只有送出帳號，但沒有送出密碼。所以沒有加密的問題 ■ *Kerberos 使用 secret key 密鑰加密消息，為身份驗證流量提供保護 ◎ Kerberos 強調存取任何服務都必須有該服務的服務票券 (Service Ticket)。要使用 File Server 的資源，必須先取得 TGS 所發放，能存取該 File Server 的 Service Ticket。要存取 TGS 前，必須先取得 AS 所發放的 Ticket-Granting Ticket (TGT)。 ○ Kerberos requires security domains, trusts, cryptography, and so forth. It works primarily in the setting of LAN. 不適用於支持 HTML、HTTP 和 HTTPS 的瀏覽器 ○ *Kerberos 建立在對稱密鑰加密技術之上，需要受信任的第三方，並且可以選擇在某些身份驗證階段使用 public-key cryptography 技術；Session key 是加密的，且需要另一個密鑰來解密它→The client' s secret key to 解密 ciphertext 密文 ○ *Kerberos 資安議題 ■ KDC 如果單點故障受到破壞，可能會導致問題，因為密鑰 secret key(對稱式)存儲在 KDC 上

	■ 攻擊者可以冒充任何合法使用者 ■ 與許多身份驗證方法一樣，Kerberos 容易受到密碼猜測的影響 ■ Golden Ticket Attack 萬能票證攻擊，這種攻擊要求攻擊者獲取金鑰分發中心（KDC）密鑰（krbtgt hash），使攻擊者可以產生「萬能票證（Golden Ticket）」，進而獲得對整個企業環境的全面存取權 ■ 應刪除「孤兒帳號（orphaned）」及不必要的特權帳號、削弱攻擊者橫向移動的能力(包括雜湊傳遞（Pass-the-Hash）、Overpass-theHash 及票證傳遞（Pass-the-Ticket）、橫向移動的一個常見罪魁禍首是本機管理者帳號) ■ Kerberoasting 攻擊依賴於收集的 TGS tickets。通過票證攻擊依賴於從 lSasS 過程中收穫的票證
CVE CVSS CWE Q: 要跟SCAP一起念	①. Common Vulnerabilities and Exposures , CVE：屬已知弱點，針對特定廠商或型號；描述安全漏洞的命名系統 ②. Common Vulnerability Scoring System , CVSS： 分數曝險值，提供一個描述安全漏洞嚴重性標準化評分系統，考量 base，temporal(時間)， environmental metric groups ③. Common Weakness Enumeration ，CWE：屬於設計階段風險審查。未針對特定廠商或型號。
滲透測試	利用漏洞進行入侵可行性測試，從攻擊者的角度來評估 工具：Metasploit 主要步驟 ①. Information Gathering，資訊收集 ②. Scanning and Reconnaissance，掃描和偵察 ③. Fingerprinting and Enumeration，指紋和枚舉 ✧ 指紋(Fingerprint)是收集各式的微物跡證(類似指紋的效果)來識別一個實體，可用在網路行銷及滲透測試。進行網站滲透測試時，會透過 HTTP header、cookie、banner 或 ICMP 的回應來識別系統， 以進一步找出該系統的弱點。 ✧ 枚舉（Enumeration）： 枚舉的主要目的在搜尋系統中共有資源與使用者資訊，以便尋找漏洞進行進一步的入侵行為。 ④. Vulnerability Assessment，脆弱性評估 ⑤. Exploit Research and Verification，利用研究和驗證 ⑥. Reporting 報告
SCAP	Security Content Automation Protocol：自動化修補協議，弱點的修補可以透過 SCAP 來作自動化的大量部署
稽核	稽核強調獨立性，獨立單位作的安全評鑑才能稱為稽核， 採用的方法都是查驗， 訪談， 測試 稽核分類： ①. 第一方(組織內部的稽核部門) ②. 第二方(客戶行使合約中所保留的稽核權，Audit Right 稽核權) ③. 第三方(四大會計師事務所，ISO 的驗證機構如 SGS， BSI， TUV，TCIC 等， 以及主管機關
Change Management	Baseline（核定過的都是基準，需要調整要經過變更申請）：範圍、時間、

	成本、組態、資安基準 Request for change→Review and Evaluation→Approve/Reject Decision→Updates to Change Log →Implementation of Approved change → Monitoring and communication
IDS IPS	IDS：只偵測、只聽、只判斷，不干涉入侵行為 IPS：偵測、判斷，干涉入侵行為 SIEM 、EDR 要一起念
Investigation and Evidence	羅卡定律 (Locard's exchange principle)：犯罪現場調查中，行為人必然會帶走一些東西，亦會留下一些東西 Investigation Type ①. 行政調查(Administrative investigation)：行政調查是指對員工所謂的不當行為的內部調查 ②. 監管調查(Regulatory investigation)：是指由政府，監管，執法，專業或法定機構發起的正式聽證會，官方調查，檢查，詢問，法律訴訟或任何其他類似程序 ③. 民事調查(Civil Investigation)： 民事調查會發現並收集進行民事審判所需的證據 ④. 刑事偵察(Criminal Investigation)：是一門應用科學，涉及對事實的研究，然後將這些事實用於刑事審判 Burden of Proof 舉證責任 ①. Preponderance of the evidence 證據優勢(Civil Case)：不須達到明晰可信 (clear and convincing proof)，更不須達到無合理可疑 (beyond a reasonable doubt) 的程度 ②. Proof beyond a reasonable doubt (Criminal Case)：刑事被告人定罪的證明標準。指所有材料都證明被告人有罪，排除一切合理的懷疑或假設 ● Evidence type ①. Real Evidence 物證 ②. Demonstrative Evidence 展示證據/實務證據：狹義上指本身並無證明價值〔probative value〕，但可用來說明或澄清有爭議的事實問題的實物，如地圖、圖表、模型、照片等 ③. Documentary Evidence 書面證據，以書面文件為表現形式的證據(書面 O/口頭 X) ④. Testimonial Evidence 專家證據：在只有專家意見才能幫助法官或陪審團解決爭議問題的情況下，專家意見是可以被採納為證據的 Admissible Evidence 可採納的證據；指具有相關性，並且依其性質(如非傳聞、不存在不公正的偏見等)法庭或法官應予接受的證據。 ①. Relevant 相關 ②. Material 材料(有內容) ③. Competent 能力(有證據能力) Chain of Custody 物證連續保管：向法庭提交物證的人，如在毒品案件中向法庭提交麻醉品作為物證的人，必須說明從他開始保管該物證直至他向法庭提交該物證的期間，他一直保管該物證的情況
備份	Full Backup：most negligible overhead and the best efficiency in terms of restoration Differential Backup：備份跟完全備份有差異的部份 Incremental Backup：備份跟前一次有增加的部份

機房	①. 選址 Site Management：事前評估(Due Diligence)、是否有天然災害(淹水…多久一次)、綠電、當地專業人員 ②. CPTED (Crime Prevention Through Env Design):透過環境設計來犯罪防治(透過大環境特性)，如圓環，不讓車直接衝向大門 ③. 門禁管制：圍牆、旋轉門(turnstile)、陷阱(mantrap) ④. 電力設施 牆壁高度?? ✧ Power Loss ■ Fault (瞬斷) --> UPS ■ BlackOut (長時間停電) --> 發電機 ✧ Power Excess 電壓不穩 ■ Spike 瞬間飆高 ■ Surage 一直都在高檔 ✧ Power Degradation ■ Sag or dip 電壓太低供電不足 ■ Brownout ： 長時間電壓過低 ⑤. 消防 ✧ 火的成因有三個要素(燃點，燃料，助燃)，事前消除這三者就可以預防， 事中由這三點可作好偵測及消防 ✧ 偵測器的類型：溫度感應、偵煙、偵熱 (差動) ✧ 抑制 (針對火災三要素) 消防系統 ✦ Water Suppression System:不讓消防水結冰 ✦ Gas 消防的化學物質:CO2 Halon 海龍(已禁止用)，FM 200 海龍替代品 ✧ 滅火器 ✦ A (Ash 灰)：東西燒掉變成灰 ，針對 一般可燃物 ✦ B (Barrel 油桶);針對油，可燃液體 ✦ C (Current 電流)；電器類火災 ✦ D (D 金屬化學物質)：金屬化學物質 ✦ K (Kitchen 廚房)：廚房 ⑥. 玻璃：人生安全考量，大片落地窗不能破人，防窺、防彈、防傷人 ✧ Laminated Glass 膠合玻璃，用在路邊大的櫥窗，碎掉也不會飛濺 ✧ Bullet-Proof Glass 防彈玻璃 ✧ Tempered Glass 強化玻璃：各種要求苛刻的應用，包括乘用車窗戶、淋浴門、建築玻璃門和桌子、冰箱托盤、作為防彈玻璃的組成部分、用於潛水面罩以及各種類型的盤子和炊具
系統開發 SDLC	System Development Life Cycle (NIST) ● Initiation，啟動 ➢ RMF-Categorize Systems：依據 impact 程度分高中低 ➢ BIA ➢ PIA 隱私衝擊評估 (Privacy Impact Assessment) ● Development/Acquisition，開發 Make/收購 Buy ➢ RMF-Select Controls：依等級進行項次的控制 ● Implementation/Assessment，實施/評估 ➢ RMF-Implement Controls：進行控制 ➢ RMF-Assess Controls：評鑑有效性

	✧ Verification，內驗，內部驗證正確性 ✧ 認證(Accreditation)：指主管機關對某實驗室或驗證機構給予正式認可（授權），證明其有能力執行某特定工作。例如：SGS 的食品服務部是經過衛生福利部 TFDA 認證。 ➢ RMF-Authorize Systems：授權系統上線 ✧ Validation，外確，外部確認有效性 ✧ 驗證(Certification):由中立之第三者確認某一項產品、過程或服務，能否符合規定要求達到一定標準。 ● Operations and Maintenance，運營和維護 ➢ Review 操作準備狀態 ➢ RMF-Monitoring Control：持續監測 ➢ 變更管理 ● Disposal，除役 ➢ 資料殘留 ➢ 資料保存政策 ➢ 資料消毒 Sanitization ✧ Clear 擦除，人員/實驗室可恢復 ✧ Purge 清除，廠商提供的工具，需送實驗室才能救回 ✧ Destroy 銷毀，物理性破壞，如碎化
軟體開發 SDLC	Software Development Life Cycle ● Planning Phase（計畫階段） ➢ 寫計畫書 ➢ 決定開發方法 ✧ Predictive Approaches 預測的方法： Plan-driven 、Formal→Waterfall 瀑布式（Plan-driven） / Cleanroom（Formal）無塵室 ✧ Iterative 反覆式（大瀑布拆成小瀑布，分期）→ Spiral Model 螺旋 ✧ Agile（Iterative 週期反覆（SDLC） + Incremental 價值見證(Value)） ■ Scrum、Kanbam、eXtreme Programming(Test-driven development 先寫測試再開發/Pair Programming 結對開發/Continuous Integration 持續整合) ➢ 組建團隊（IPT，Integrated Product Team 整合產品團隊） ✧ DevOPs：(傳統) Dev、IT、QA ✧ DevSecOPs： Dev、IT、QA、Security ✧ CI （Continuous Integration)： 連續整合、整合測試 ✧ CD（Continuous Deployment）交付又安裝 ● Analysis Phase 步驟 ➢ 分析的分是【分解】，析則是指【深入了解】 ➢ 需要（Need)：利害關係人腦袋中的東西 ➢ 需求（Requirement)：寫下來並進行管理的需要 ➢ Elicitation 引出/收集 ➢ Analysis 分析 ➢ Specification 記錄規格：USR、SRS、User Case、User story(便利貼)

	➢ Verification & Validation 內驗外確(整個 SLDC 都會有) ➢ Change Management ● Design Phase ➢ 設計紙上談兵的解決方案（通常都是設計圖） ➢ 架構設計(主要元素及其關係)、細步設計、設計審查 ➢ 設計審查 ✧ Verification：內驗正確性 ✧ Validation：外確有效性 ✧ 威脅塑模 ● Development Phase ➢ Software Environment：Development、Testing、Staging、Production ➢ OOP：Object-oriented programming 物件導向程式設計 ✧ 封裝（Encapsulation)：將物件內部的資料隱藏起來，只保留特定的方法 Interface 與外界聯絡 ✧ 繼承（Inheritance)：透過繼承機制將共通的資料(屬性)、方法寫在父類中，而繼承的稱為子類別。子類別繼承了父類別。例如：計程車(子類別)繼承了汽車(父類別)原有的屬性以及方法，也新增了自己特有的屬性(driverName) ✧ 多形 Polymorphism：簡單來說就是相同名稱的方法(Method)，多個相同名稱的方法，傳入不同的參數，會執行不同的敘述。 ■ 多載(Overloading) — 是指說在相同類別中，定義名稱相同，但是參數個數不同，或是參數型態不同的函式，這樣就可以利用參數個數或者參數型態，呼叫到對應的方法。例如：一個計算面積的方法，如果傳入一個參數，就當正方形來算面積；傳入兩個參數，就當成長方形來算面積。 ■ 複寫(Overriding) — 是指覆寫掉父類別中的函式。例如：動物類別(父類別) getLegs()方法被鳥類別(子類別)覆蓋。 ➢ 版本庫 ● Testing Phase Types of Testing 測試類型 ● Maintenance ○ 上線維運階段（職責分離）
威脅塑模 Threat Modeling	風險識別：看圖找問題，尤其是資料流程圖(DFD)；找到的威脅還要根據 STRIDE 來進行分類，步驟如下： ➢ Diagram Application Architecture，把圖拿出來 ➢ Identify Threats，看圖(如 Data Flow Diagram) ➢ Threat Categorization - STRIDE 是從攻擊者的角度，把威脅劃分成 6 個類別 風險分析：根據 DREAD 來計算每個威脅的風險分數 ➢ Identify ， Prioritize & Implement Controls，處理問題 ➢ 風險分析、風險評估、風險處理 ➢ Threat Analysis - DREAD 評估風險敞口

	■ Damage Potential 潛在損害：如果利用漏洞造成的損害有多大 ■ Reproducibility 重現性 ： 重複產生攻擊的難度有多大 ■ Exploitability 可利用性 ： 發起攻擊的難度有多大 ■ Affected users 受影響用戶 ■ Discoverability 威脅是否容易探測 風險評估: 這點沒有特別強調 風險處置: 要針對威脅提出處置方式, 評估殘餘風險及文件化 Document & Validate，紀錄他
Types of Testing 測試類型	①. Black vs White Testing 黑箱測試(Basic testing)：測試者對於受測標的物一無所知 ➢ 隨機測試(Random testing) 是一種黑箱測試，通過生成隨機的獨立輸入來測試程序 ➢ 猴子測試，用戶可以通過提供隨機輸入並檢查行為或查看應用程序或系統是否崩潰來測試應用程序或系統 白箱測試(Comprehensive testing)：測試者對於受測標的物都知道 灰箱測試(Focused testing)：介於黑白之間 *補充 In blind testing or double-blind testing, **passive testing** (information gathering) is usually conducted before active testing to keep the testing unaware or in secret. ②. Active vs Passive Testing ➢ 互動：Interaction(網路/主動) vs non-Interaction(竊聽/被動) ➢ Passive：被動、弱掃、Sniffing 網路監聽(測試者與測試標的物之間沒有直接的互動) ➢ Active：honey pot，Decetion technology，威脅狩獵 (Threat Hunting)、Ping(測試者與測試標的物之間有直接的互動) ➢ Offensive 進攻：Botnet takedown 殭屍網路，Hack-back attack ③. Manual vs Automated Testing: Tester vs Test Runner/ Harness (Test Runner/ Harness) ④. Static vs Dynamic Testing (Software) ➢ 靜態測試：未執行該軟體（屬 Program），放置於硬碟進行測試(源碼掃描) ➢ 動態測試：已執行該軟體（屬 Process），已被載入記憶體中進行測試(壓力測試)
Software Testing Techniques	①. Software Testing Techniques：Unit Testing (+TDD (測試先行)：先寫單元測試，再寫程式) ②. Code Review： ⊥ pair programming 結對編程（同儕）(Agile)，最高檔 code Review，即時檢視 ⊥ 提交審查（主管） ⊥ 機器人掃描 Code Review ⊥ 正式審查檢視（專家）：費根審查 Fagan Inspection ③. Integration Testing 整合測試，與其他人的 code 合併後，基本（常見）方式，將所有人的 Unit Testing，再跑一次 ⊥ 1 Step：程式和單元測試都編譯一次 ⊥ 2 Step：將所有人的 Unit Testing＋單元測試，再跑一次

	④. Regression Testing 回歸測試，測到沒問題;重複執行以前的全部或部分相同的測試工作，將結果與早期版本的軟件產生的結果進行比較 ⑤. Interface Testing，介面測試（UI 測試、API 測試） ⊥ Fuzz Testing，模糊測試，用工具（Fuzzey）隨機動態，產生測試資料，透過使用者或是 API 介面進行測試（Interface Testing)。容易破壞語意完整性，它向測試的程序提供無效的輸入，zzuf 是最著名的模糊器之一 ⑥. Misuse Case Testing，誤用測試，一種使用實現者不期望的功能的方法，允許攻擊者根據攻擊者的動作（或輸入）影響功能或使用功能的結果 ⑦. Stree Testing，壓力測試，測試系統上限 ⑧. Performance Testing，效能測試，在一段壓力下，測試效能狀況 ⑨. Security Testing，滲透、弱掃 ⑩. User Acceptance Testing（UAT)，驗收測試 ⑪. Installation Testing，安裝測試 ⑫. Synthetic Transactions 合成交易，寫腳本進行監控，這些腳本或工具可模擬通常在應用程序中執行的活動，這種類型的測試或監視通常與定制開發的 Web 應用程序關聯 ⑬. Synthetic monitoring 合成監測，是一種主動的網站“監視”技術，通過在網絡瀏覽器中部署行為腳本來模擬真實客戶（或最終用戶）通過網站的路徑來完成。Synthetic monitorin 對於高流量站點（例如電子商務）在發布之前進行測試至關重要 ⑭. Test Coverage Analysis，測試覆蓋率分析，重點是測試的分母為何；描述程式中原始碼被測試的比例和程度，所得比例稱為代碼覆蓋率 ⊥ Expressions and decision structures 軟件測試覆蓋率分析方面具有最精細的測試 ⊥ 在軟體開發項目中使用開源組件，最不關心的是測試覆蓋率 ⊥ **原始碼行數（Source lines of code，LOC)，用來估計程式開發生產力或可維護性** ⑮.「入侵與攻擊模擬」(Breach and Attack Simulation, BAS）自動模擬駭客進行多面向攻擊的一種工具，此種工具之目的在驗證企業的資安防護能力；結合紅隊（攻擊）和藍隊（防禦）技術與自動化的系統，以在針對環境運行時模擬 advanced persistent threats ⑯. DAST：動態應用程式安全測試（Dynamic Application Security Testing）技術在測試或運行階段分析應用程式的動態運行狀態。它模擬黑客行為對應用程式進行動態攻擊，分析應用程式的反應，從而確定該 Web 應用是否易受攻擊，如 fuzzing and web application vulnerability scanning ⑰. SAST：靜態應用程序安全測試（static application security testing)，Code reviews and static analysis packages analyze the code itself but do not execute it
關聯式資料庫	①. 關聯 Relation 就是 Table 本身 ②. 欄（Column / Field → Attribute) ③. 列（Row / Record → Tuple) ④. 有幾欄 稱為 degree

	⑤. 有幾筆 Record 稱為 cardinality
	實體完整性 Entity Integrity：Primary Key ➢ 實體完整性出問題（資料重複） ➢ Primary Key 唯一識別 參考完整性 Referential Integrity ➢ 參考完整性出問題：Foreign Key 找不到源頭 ➢ orphan record:Foreign key 指向已經不存在的資料 ➢ 定義 Foreign Key，建立 Relation ➢ 交易控制：Clark-Wilson Model 語意完整性 Semantic Integrity (Domain Integrity)：輸入資料型態或是範圍（Domain）不對，可透過欄位型態指定或是長度進行限制
Aggregation and Inference	資料庫聚合 Aggregation:是指個別資料雖不具機密性，但聚合數筆資料後卻可獲得機密資訊 ➢ 尤其是關係數據庫，SQL，Excel 和其他軟件提供了所謂的集合函數，例如 Sum ( )，Average ( )，Count ( )，Max ( ) 和 Min ( ) 等。這些函數正在處理“一組數據，”而不是“單個數據記錄”，因此由於“聚合”而導致數據洩露 資料庫推論 Inference：從已儲存的資料庫中推論出新的資訊，以找出原本未知的資訊，例如從 Birthday 推算出 Age
Polyinstantiation 多實例化	有真有假的資料混合，避免猜測攻擊 Database developers use polyinstantiation, the creation of multiple records that seem to have the same primary key, to protect against inference attacks(推論 Inference 攻擊)
惡意軟體	①. 病毒：電腦病毒將自身附加在程式或檔案上，以便從一台電腦傳播到另一台電腦，並隨著它所到之處進行感染。若無人為操作（例如執行被感染的程式），病毒本身是無法散播的 ②. 蠕蟲：無需人為操作，就有以自行傳播的能力，利用您系統上的檔案或資訊傳遞功能，使其得以自主散播。如，蠕蟲會向您通訊錄中的每一個人傳送其自身的複製（郵件） ③. 特洛伊木馬：不是病毒，它是一種看起來像正常應用程式的破壞性程式。與病毒不同，特洛伊木馬程式不會自我複製，但是同樣具有破壞性。留下一個後門入口，讓惡意使用者/程式存取您的系統，以盜取機密或個人資訊 ④. Buzzwords 騙點擊 ➢ Pranks：Fun Viruses 惡作劇 ➢ Hoaxes：Rumor 謠言 ⑤. Backdoor 後門 ➢ Trapdoor / Maintenance Hooks ➢ Logic Bombs ⑥. Botnets 殭屍網路是指駭客利用自己編寫的分散式阻斷服務攻擊程式將數萬個淪陷的機器 ➢ Bots 殭屍電腦（robots)：Controlled agent ➢ Zombies：Bod bots ➢ Bots Master：The Person in Command and Control （C&C or C2） over the botnets
Layer Vs. Tier	Layer（邏輯上）三層式（處處都安全的考量）

	➢ Presentation Layer（畫面呈現、收資料、輸出入介面）: 容易受到 SQL Injection、XSS、Buffer Overflow 攻擊 ➢ Business Logic Layer：業務邏輯層中的處理可能容易受到攻擊，但需要動態測試。專注於數據流的威脅建模基本上是靜態審查，因此不如動態測試有效。因此，在進行威脅建模時，業務邏輯層中的處理是最不關心的問題 ➢ Data Access Layer： ● 資安考量 ○ 實體完整性問題 ■ 資料重複就破壞完整性 ■ Primary Key 就是用來保護資料完整性 ○ 參考完整性問題 ■ 查表動作 叫做 Reference 若無資料（斷頭）稱為 Reference 完整性出問題 ■ 用交易 Transaction 來保護參考完整性 （透過交易去控制 Clark-Wilson Model） ● SOD 職責分離 ● 交易 ● TP/IP/CDI ■ 避免：主表格與明細表格不能出現斷頭 ○ 語意完整性問題 ■ 格式 、範圍 … 解讀上的意義 無意義 ■ 控制資料型態 控制資料範圍 限定輸入方式（改下拉式） Tier（實體） ➢ 放在不同的電腦上 ➢ 與程式部署架構有關
RESTful Architecture	RESTful Style：調用服務系統時，使用 HTTP Verb。RESTful API 使用 標準 HTTP 方法來操作數據。CRUD Create：POST-Insert Read：GET-Query Update：PUT-update Delete：Delete
SOA，Service-Oriented Architecture	協定：SOAP（XML） 軟體服務化/服務即程式 ➢ 服務對象是其他程式 ➢ API 提供名稱，輸入參數，返回值 ➢ 第一代：WebService ➢ 第二代：Microservice Roles ➢ Service Registry，服務註冊 ➢ Service Provider，服務提供 ➢ Service Consumer，服務使用者 Implementations，實現 ➢ Web Services Description Language（WSDL）描述 Web 服務發布的 XML 格式

	➢ Universal Description Discovery and Integration（UDDI）基於 XML 的跨平台的描述規範，可以使世界範圍內的企業在網際網路上發布自己所提供的服務 ➢ Simple Object Access Protocol（SOAP）交換資料的一種協定規範，使用在電腦網路 Web 服務（web service）中，交換帶結構的資訊（XML）
MicroServices Architecture 微服務	● 外部服務內部化，單一職責原則 Single responsibility principle ● 是一種低耦合架構，可以通過重新建構單一應用程式來實現，即將應用程式元件轉變為自成一體的網路服務，適合部署在可擴展或彈性容器或無伺服器環境中。 ● API Gateway：微服務對外通訊(Client or System) ➢ 確保合理的請求(request)率 ➢ 將來自舊客戶端的請求重定向到新版本的服務 ➢ 避免級聯故障(cascaded failure)的可能性 ● Service Mesh 服務網路 : Service Mesh 用於管控微服務架構上的一群服務，它並不是服務本身，怎麼管理或追蹤成群的服務行為才是它所在乎的，「管理服務之間的網」或是「追蹤服務的行為」，例如：**服務與服務之間的連線** service-to-service communication 是否需要引入 Service Mesh： ➢ 運行敏感（sensitive）的服務 ➢ 運行無法信任（untrust）的工作 ➢ 運行多租戶（multi-tenant）的工作 ➢ 需要讓系統具備及強化可觀察性、控制及網路通訊安全，就需要引入 Service Mesh，如內部服務與服務之間的連線加密，就是相當常見的場景
Container Orchestration 調度	①. 容器服務是一種程式運行的虛擬化環境，屬作業系統 OS 層級 ②. 提供一個獨立與封閉的電腦資源，例：CPU、記憶體與應用程式組成要件（應用程式碼、依賴度）的虛擬環境。每個儲存裝置如同一個獨立的套件以執行應用程式或相關服務，無須再仰賴於底層的計算環境。 ③. 執行速度快，儲存空間小，資源有效運用 ④. Google K8sKubernetes（K8s）是用於自動部署、擴充和管理「容器化（containerized）應用程式」的開源系統 ⑤. docker（Swarm）Docker 公司所推出原生的容器調度管理平台 ⑥. RedHat OpenShift 是 PaaS 雲端運算平台，供使用者建立網路應用（App、網站） ⑦. Container 的不變性意味著“隨時間不變或無法改變”。我們可以將容器視為只讀的；更新容器的唯一方法是用新容器替換它。無狀態意味著容器不會在其本地存儲中保留數據；但是，遠程存儲庫用於跨容器持久化和共享狀態。 ⑧. 容器不需要管理程序來支持應用程序虛擬化。容器可以部署到裸機，無需 Hypervisor 管理的虛擬機 ⑨. Hypervisor 提供比容器更高級別的隔離 Isolation，因為容器是共用作業系統 OS 層級虛擬化技術 ⑩. 如果程式佈署在二台 VM，跟一台機器 2 個容器，隔離的角度 VM 比較好，因為容器是共用作業系統 OS 層級虛擬化技術，安全性比較高 ⑪. Linux 主機只能運行為 Linux 構建的容器，Windows 主機只能運行

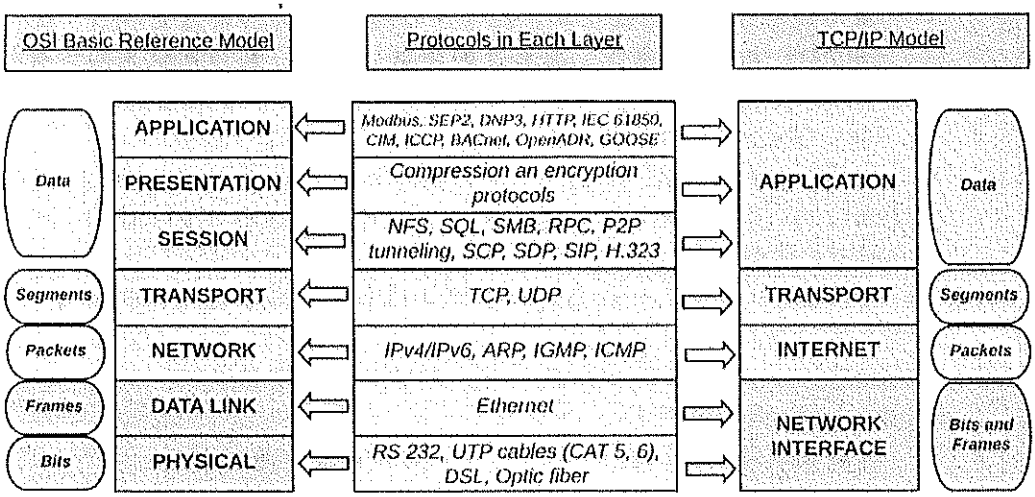
	Windows 容器 ⑫. 容器技術的引入可能會破壞組織內現有的文化和軟體開發方法。傳統的開發實踐、修補技術和系統升級過程可能無法直接應用於容器化環境，員工願意適應新模式很重要

Hypervisor	Hypervisor，又稱虛擬機器監視器（Virtual machine monitor，VMM），是用來建立與執行虛擬機器的軟體、韌體或硬體。
	類型 I：原生或裸機 hypervisor 虛擬機器監視器
	- 這些虛擬機器管理程式直接執行在宿主機的硬體上來控制硬體和管理客作業系統。
	- 特點：需要硬體支援、虛擬機器監視器作為主作業系統、執行效率高
	- 舉例
	  ✧ VMware ESX 伺服器版本
	  ✧ Xen 3.0 及以後版本
	  ✧ Virtual PC 2005
	  ✧ KVM
	● 類型 II：寄居或代管 hypervisor，這些虛擬機器管理程式執行在傳統的作業系統上，就像其他電腦程式那樣執行。

Cloud Computing 運算資源服務化	● Essential Characteristics 基本特徵（NIST SP 800-145） ➢ On-demand self-service：隨選自助服務 ➢ Broad network access：寬頻連接 ➢ Resource pooling：資源池 ➢ Rapid elasticity：快速彈性 ➢ Measured Service：可量測 ● ISO 17888 ➢ Multi-Tenancy：多租戶（SaaS） ● Service Models ➢ Software as a Service (SaaS)：使用軟體（服務），直接使用系統，一般使用者，雲端的共享責任：Data，用戶通常使用精簡用戶端經由一個網頁瀏覽器來存取軟體即服務 ➢ Platform as a Service (PaaS)：程式平台，程式上傳，即可使用，程式設計師，雲端的共享責任：Application，提供運算平台與解決方案服務 ➢ Infrastructure as a Service (IaaS)：VM，自己裝作業系統，系統工程師，雲端的共享責任：OS ● Deployment Modes ➢ Private Cloud：自己人使用 ➢ Public Cloud：公眾使用 ➢ Community Cloud：特定利害關係人一起使用 ➢ Hybrid Cloud：上述情況混用
Cloud Computing Role	Cloud Carrier Cloud Service Provider，CSP Cloud Broker Cloud Consume Cloud Auditor
CSA STAR	STAR 認證 = ISO27001 + 雲端控制矩陣 +成熟度評估

	Level 1：自我評鑑(Self Assessment)，由雲端服務提供商填寫自我評鑑問卷，自我宣告對於 CCM 的遵循程度，並上傳至 CSA 的官方網站供公眾查閱。 Level 2：驗證(Certification)，由第三方機構依據 ISO 27001、CCM 及管理能力模型進行評鑑後，授與管理能力等級獎牌的一系列活動。 Level 3：持續監視(Continuous)，仍在發展中，主要精神為使用即時監視的技術，持續收集稽核證據，以確認對於顧客要求的滿足程度的方案。

#	名稱	關鍵	資安關注議題
7	Application Layer	應用程式	FTP 明文傳送、DNS cache poisoning(DNS spoofing)、流氓 DHCP
6	Presentation Layer	編碼 / 壓縮 / 加解密	Brute Force，Meet in the middle(2DES)，Frequency Analysis，Birthday Attack，彩虹表
5	Session Layer	全雙工/半雙工/單工	Access Token 劫持 (DNS Spoofing、中間人、XSS 跨站攻擊-違反同源政策)
4	Transport Layer	保證送達、不保證送達	UDP：Fraggle attack ； TCP：SYN Flood Attack ， 聖誕樹攻擊
3	Network Layer	(定址)Addressing 、(選徑)Routing	Ping of Death ， Smurf Attack ，TTL fingerprinting / footprinting，Teardrop 攻擊，流氓路由器
2	Data Link Layer	Media Access Control，MAC 、Logic Link Control，LLC	ARP cache poisoning、MAC flooding
1	Physical	線材、接頭、訊號、傳輸方式	訊號衰減、竊聽、破壞、干擾



#	內容
Wi-Fi	①. 當一台裝置想要連上一個 Wi-Fi 網路的時候，會經過幾個步驟 ➢ 掃描 Probe Request ➢ 認證 Authentication request ➢ 關聯 Association request ➢ 四向交握 4-way Handshake ➢ Group key handshake ②. ad-hoc mode & Infrastructure mode ➢ ad-hoc mode 點對點模式：簡單的說，兩台電腦之間，直接透過無線網路卡就可以上網，而不用使用 AP。讓兩個裝置互通的一種模式，若兩裝置都沒連上 internet，只靠 ad hoc 模式就單單只能讓兩裝置互通而已，以達到資源共享(印表機 、檔案、網際網路等) 。 ➢ Infrastructure mode 基礎建設模式：所有的設備皆需連接到一個存取點(如 Access Point；AP)，透過存取點來連接其他的無線網路設備，或是存取有線網路的資源 。 ③. 無線網路 Media Access Control(MAC) ➢ 訪問機制走 CSMA/CA （載波偵測多重存取/碰撞避免） ➢ 通訊協定 IEEE 802.11
Wireless Security (Client to wireless access point)	①. 第 1 代 Wireless Security (Client to wireless access point) ✧ 802.11 ✧ WEP (Wired Equivalent Privacy)已被破解 ■ 使用 RC4 ■ 因為美國政府管制長度，導致被破解 (僅 64 Bit / 128 Bit)，金鑰長度太短 ■ IV 24 Bits ②. 第 1.5 代：WPA 使用 TKIP(可用韌體升級)，使用 RC4 作為底層密碼，確保機密性 ③. 第 2 代： ✧ WPA2 在 CCM 模式下使用 AES，一種分組密碼 (CBC-MAC 計數器) ✧ WPA2 是 Wi-Fi 的安全機制，可進行連接無線網路時的身份驗證，

	強化機密性、完整性及真實性等。WPA2 使用的加密器是 AES，在 Counter mode 下運作；真實性則是以 CBC-MAC 來計算訊息驗證代碼(MAC)。Counter 加上 CBC-MAC 合稱 CCMP(首字分別是 Counter，Cbc，Mac， Protocol) ④. 第 3 代:WPA3 使用 HMAC 雜湊訊息鑑別碼（Hash-based message authentication code）來確保真實性；不可否認性由 Digital Signature 強制執行
Wi-Fi 攻擊	①. War Driving：開車繞一圈蒐集有多少基地台，基地台使用哪種協定，是否使用預設密碼 ②. WarChalking：將蒐集來的做記號(Ex.阿里巴巴做記號) ③. Replay attack 重放攻擊 ➢ 惡意或欺詐的攔截有效資料，並將其在網路中重複傳輸 ➢ 解決方法: ✧ 序號：通訊對象必須記錄最後一個序號 ✧ 時間戳：可能有時間差被拿來利用 ✧ nonce：每次通訊都產生隨機值，通訊數據量也會增大 ④. Evil Twin（完全模仿你，訊號強就會連結過來），如公共基地台（用相同的 SSID 與相同金鑰） ⑤. 藍芽攻擊 竊聽、訊號側錄（中間人攻擊）、攔截或重導兩個藍牙裝置之間的通訊 ⑥. Control zone(阻止訊號外洩) against Tempest(風暴) ✧ Faraday cage 鐵絲網 ✧ White Noise ✧ Wire-meshed space 線網空間
破密攻擊	①. Ciphertext Only Attack(COA)只有截到密文：指的是在僅知已加密文字的情況下進行攻擊，此方案可同時用於攻擊對稱密碼體制和非對稱密碼體制 ②. Know Plaintext Attack(KPA)已知明文攻擊：指攻擊者掌握了某段明文 x 和對應密文 y，透過這個組合去了解演算法的內容和 key ✧ 已知部分 123 加密成 xyz ✧ 電子密碼本（Electronic codebook，ECB）有 pattern ✧ 密碼區塊連結（Cipher-block chaining，CBC ）則有 IV ③. Chosen Plaintext Attack(CPA) 選擇明文攻擊：是指一次丟好幾組的明文到演算法中，去觀察其密文的變化進而找到 key(針對非對稱) ④. Chosen Ciphertext Attack(CCA) 選擇密文攻擊：已經知道密文轉回明文的方法，以自選的一組密文來推出明文是什麼。(自選一組密文給對方解，就知道對方用什麼金鑰) ⑤. Brute-force attack：暴力攻擊，是一種密碼分析的方法，即逐個測試可能的密碼，直到找出真正的密碼為止(全部 try 一次) ⑥. 彩虹表攻擊(MD5)：是一個用於加密雜湊函式逆運算的預先計算好的表，常用於破解加密過的密碼雜湊，如 MD5，使用加鹽的金鑰衍生函式可以使這種攻擊難以實現。彩虹表常常用於破解長度固定且包含的字元範圍固定的密碼（如信用卡、數字等）。這是以空間換時間的典型實踐，比暴力破解（Brute-force attack）使用的時間更少，空間更多，使用加鹽的金鑰衍生函式可以使這種攻擊難以實現 ⑦. Access control 攻擊：Password attack、Dictionary attack、生日

	攻擊 ⑧. Quantum Computing：量子電腦是新一代的機器，極有潛力能運用 AI 技術，迅速解決傳統電腦需要很長時間才能排除的問題
Cryptographic Vulnerabilities-Algorithm Vulnerabilities	①. Analytic Attack (Algorithm) ②. Frequency Analysis (Classical ciphers)：字母出現的機率並非完全相同，適用於破解凱薩密碼 (Caesar cipher) ③. Birthday Attack (Hash)：生日攻擊 (機率分布)，碰撞(不同值但雜湊相同，不同密碼都能登入) ④. "Meet" in the Middle (2DES)：概念上是把一系列的計算分成前半部份和後半部份，利用建立表格的方式，找到前半部份計算與後半部份計算相等的地方-->3DES ⑤. 中間人攻擊 (Man-in-the-middle attack，MITM) 在密碼學和電腦安全領域中是指攻擊者與通訊的兩端分別建立獨立的聯絡，並交換其所收到的資料，使通訊的兩端認為他們正在通過一個私密的連接與對方直接對話，但事實上整個對談都被攻擊者完全控制
Cryptographic Vulnerabilities-Entropy Vulnerabilities 熵 (加密演算法的弱點)	利用計算機硬件或操作系統的弱點 ➢ 補充熵的意思 ✧ 混亂程度 ✧ 越亂越不容易被猜到 ✧ 越沒有規則可循 ✧ 金鑰也就不容易被取得
Data Link layer 攻擊	①. ARP(Address Resolution Protocol 位址解析協定) cache poisoning，負責將 IP 位址轉換成 MAC 位址的一種通訊協定 ➢ ARP cache：為了讓網路運作有效率，所以在各裝置上都會快取 MAC 與 IP 對應 ➢ ARP spoofing 欺騙 / Poisoning 中毒：發送一個假的 ARP 封包竄改 ARP Cache，使得資料無法正確傳輸到目的地，造成網路無法連結 ➢ RARP(Reverse Address Resolution Protocol)，主機電腦用自己硬體位址 (MAC 位址) 向伺服器詢問自己的 IP 位址 ②. MAC Spoofing 欺騙：攻擊者向 Switch 提供虛假的 MAC Address 讓它寫進 MAC Address Table 之中，所有要傳給 Host 的訊息，都會被誤傳至偽冒的攻擊者 ③. MAC Flooding 洪水 ➢ Switch 的 MAC Address Table 被虛假的紀錄塞滿，正確的紀錄就會無法寫入，導致 無法找出目的地 Port，於是改為用 Broadcast 傳送資料。Broadcast 會佔用網絡頻寬，而最嚴重的問題是資訊會被暴露於整個網絡之中，被攻擊者擷取 ➢ Switch 原本能過濾封包，Buffer 灌爆後自動降級為 Hub，不需要 mirror ， 直接能 sniffer ④. DHCP Spoofing：屬於 DDoS 攻擊，使 DHCP 伺服器沒有可分配的 DHCP 地址，造成 DHCP 位址集區枯竭，從而使網路內正常主機無可分配的 IP 地址。同時，駭客利用冒充 DHCP 伺服器，為使用者指派一個經過修改的 DNS 伺服器位址，引導至預先配置好的假的金融網站或電子商務網站，騙取使用者的帳戶和密碼，這種攻擊的危害是很大 ⑤. Virtual LAN Hopping 跳躍：VLAN 本身不足以保護環境的安全，惡意

	黑客通過 VLAN 跳躍攻擊，即使未經授權，也可以從一個 VLAN 跳到另一個 VLAN ⑥. CAM Table Overflows：目的是透過發送大量偽造 MAC address 的封包，使得 Switch 的 CAM table 紀錄的對應資料中不含有正常的主機 MAC address，在這種狀況下 Switch 的工作方式就如同集線器一樣，將收到的封包將所有的連接埠送(除了來源的連接埠)，攻擊者便可藉此竊聽所有流經交換器的訊息 ⑦. Spanning Tree Protocol Attacks ：使 Switch 無法正常運作 ⑧. CDP(Cisco Discovery Protocol)/LLDP (Link Layer Discovery Protocol) Reconnaissance 偵查：是 Cisco 獨家的 Protocol，只可在 Cisco 產品上執行。透過 CDP，網管人員可以查看該設備的物理連接，得到相鄰設備的資訊，例如：型號、IOS 版本等等。現在一般都會使用 CDP Version 2，而 Cisco 設備預設亦會是 Version 2。
Network Layer 攻擊	①. Teardrop 攻擊(IP)：會傳送重疊、過大而超載的封包至目標機器，可能造成例外 OS 藍屏或提權 ②. 流氓路由器：廣播錯誤的路徑，影響流向，BGP 劫持，影響機密性、可用性 ③. ICMP 協定 ➢ Ping of Death (ICMP)：透過送出大型的封包 (長度大於 65535 bytes) 讓受害者電腦癱瘓 ➢ Smurf DoS attack (ICMP)：主要是廣播大量的 ICMP 到網路上，並且將來源 IP address 假造為受害者電腦 ➢ TTL fingerprinting / footprinting (ICMP)：不同的設備會回復不同的 TTL(Time to live) ，可以探測操作系統，服務或程序的名稱和版本以及其他信息
Transport Layer 攻擊	①. Fraggle attack(UDP Flood DoS):類似 smurf，用 UDP protocol 送出假造來源的 UDP broadcast 封包至目標網路，以產生放大的資料流 ②. SYN Flood Attack(TCP)，是一種阻斷服務攻擊，起因於攻擊者傳送一系列的 SYN 請求到目標系統，但不用 ACK 完成連接(消耗資源) ➢ 阻止方法：SYN Cookie，縮短 server 等待 ACK 時間 ③. 聖誕樹攻擊(TCP)：將封包狀態的 Flag 全亮、亂填
Session Layer 攻擊	Access Token 劫持 ①. 中間人 ②. DNS Spoofing 是指「偽造 DNS 紀錄」，比較常聽到的是「DNS 快取污染」(DNS Cache Poisoning) ③. XSS 跨站攻擊
Presentation Layer 攻擊	①. 破密分析或攻擊 ➢ Brute Force 暴力法 ➢ Meet in the middle (密碼學上以空間換時間的一種攻擊) ，2DES 被中間再相見破解 ➢ Frequency Analysis 頻率分析 ②. 雜湊 Hash ➢ 彩虹表 ➢ Birthday Attack 生日攻擊
攻擊	①. TFTP (Trivial File Transfer Protocol) 簡單檔案傳輸協定，用在 update 韌體，不須驗證 (開機可以去下載 image)

	②. FTP： 明碼傳輸，改走 IPSec，改採 SFTP ③. DNS cache poisoning，使 DNS 查詢返回錯誤回應並將使用者導向到錯誤網站 ④. 流氓 DHCP（惡意）：偽造 Mac 地址耗竭正常的 DHCP 伺服器的 IP 位址，然後黑客用自己的主機偽造一個 DHCP 伺服器，那麼新連上內網的主機只能使用流氓 DHCP 伺服器分配的 IP，這樣黑客的主機就變成了內網網關，可以藉此控制內網中其他主機的網絡流量;微軟 AD 可以設定 DHCP Server 白名單，但 Linux 則無法管制、Cisco 的 Switch 可以鎖僅接在特定的 Port 才能配發 DHCP
DOS Attacks 影響 CIA 的 Available	①. 死亡之 Ping（Ping of death， POD），是一種向目標電腦發送錯誤封包的或惡意的 ping 指令的攻擊方式。通常，一次 ping 大小為 32 位元組（若考慮 IP 標頭則為 84 位元組）。在當時，大部分電腦無法處理大於 IPv4 最大封包大小（65，535 位元組）的 ping 封包。因此發送這樣大小的 ping 可以令目標電腦崩潰 ②. 大地攻擊 LAND 攻擊（區域網路阻斷服務攻擊，Local Area Network Denial attack），是阻斷服務攻擊（DoS 攻擊）的一種，通過傳送精心構造的、具有相同源位址和目標位址的欺騙封包，致使缺乏相應防護機制的目標裝置癱瘓。 ③. Buffer overflow attack：使資料超過了處理程式回傳堆疊位址限制的範圍時，程式出現的異常操作(XP) ④. SYN flood 或稱 SYN 洪水、SYN 洪泛是一種阻斷服務攻擊，起因於攻擊者傳送一系列的 SYN 請求到目標系統。(TCP) Transport Layer ⑤. Teardrop 攻擊是一種拒絕服務攻擊，是一種基於 UDP 的病態分片數據包的攻擊方法，英文「Tear」是「眼淚」的意思，「drop」是「掉落」的意思，顧名思義，Teardrop 攻擊是一種令人落淚的攻擊手段，可見其破壞威力很強大(IP)，Network Layer ⑥. Smurf 攻擊是一種病毒攻擊，以最初發動這種攻擊的程式 “Smurf” 來命名。這種攻擊方法結合使用了 IP 欺騙和 ICMP 回複方法使大量網路傳輸充斥目標系統，引起目標系統拒絕為正常系統進行服務(DoS)，Network Layer ⑦. Fraggle 類似於 Smurf，使用 UDP 應答消息而非 ICMP(DoS)，Transport Layer
Web 漏洞	①. SQL Injection：SQL 隱碼，輸入的字串之中夾帶 SQL 指令，在設計不良的程式當中忽略了字元檢查，那麼這些夾帶進去的惡意指令就會被資料庫伺服器誤認為是正常的 SQL 指令而執行，因此遭到破壞或是入侵(Presentation Layer) ✧ 解決方式:使用參數化查詢、執行輸入驗證、限制用戶特權 ②. XSS(Cross-Site Scripting)：允許惡意使用者將程式碼注入到網頁上，是當網站讀取時，執行攻擊者提供的程式碼。XSS 通常是透過 HTML/JavaScript 這類不在伺服器端執行、而在使用者端的瀏覽器執行。可用來竊取用戶的 cookie，甚至於冒用使用者的身份 ✧ Hacker ：將惡意的 JavaScript 片段輸入至站台（Ex.表單、留言板），此惡意片段，可偷 cookie（含 token）、資料、並送到另一個站台（B 網站）因此稱為跨站 ✧ A 網站：儲存此惡意片段 ✧ 受害者瀏覽網站：光瀏覽就有可能載入此惡意 JS

	✧ 同源政策(Same-origin policy)：A 網頁設置的 Cookie，B 網頁不能打開，除非這兩個網頁”同源”。所謂”同源”指的是”三個相同”，協議相同、域名相同、Port 相同，JS 只能跟原本的網站互動，若跨站則違反此原則 ③. CSRF(Cross Site Request Forgery，跨站請求偽造），攻擊者通過某些技術手段欺騙使用者的瀏覽器，去存取一個使用者曾經登入、已認證過的網站。This attack is triggered by the end user, not initiated by the attacker
其他	①. limit horizontal privilege escalation(橫向移動是攻擊者使用非敏感性帳戶來取得敏感性帳戶存取權的方式): 使用 MFA(Multi-factor authentication)來阻止 ②. Side-channel P.272（正常管道以外的攻擊）：針對硬體(物理 Physical)的攻擊，但無需破壞硬體，常見的是時序分析及錯誤分析(故意製造錯誤)例如：耗電 poer consumption、時間資訊 Timing informtaion、功率消耗、電磁 Electromagnetic emissions 泄露或甚是聲音 Acoustic（sound）可以提供額外的資訊來源，這可被利用於對系統的進一步破解 ③. Buffer Overflow ➢ 灌爆（塞資料+惡意程式）， ➢ 例外時 return 至特定位址（提權執行…） ➢ 進而執行惡意執行指令 ➢ Buffer：Areas of the main memory, e.g., stack or heap