20200823

Timeline

Description automatically generated

一、Incident Response

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Event : 任何已經發生的事情，不管好或是不好  發生的Event通常會accounting起來  Incident : 簡單來說就是不好的Event  IT Incident 是指 資訊系統的服務水準下降  Incident在資安裡面談的是指安全事故 security incident  安全事故需要由CISO來定義, IT事故是CIO來定義(ITIL的事故管理有詳細介紹)  **安全功能：資安要做什麼，CISO需要定義清楚範圍，在作事故規劃時界定**  Disaster災難：嚴重的事故  誰來定義災難？多嚴重算是災難是由BC或DR相關負責人來定義，視組織而定  NIST對災難的定義是： 事故嚴重到在本地無發恢復，必須要在異地進行重建  老師補充   |  | | --- | | 有關災難, 上課採用NIST的定義. 基本上, NIST的災難是指機房與設施等技術層面(Tier 3)的事故, 嚴重到必須搬家(relocation)重建, 才能稱為災難. 災難重建主要是技術的復原, 如機房或設施的重建, 讓資訊服務可以恢復. 在業務面的流程都必須復原, 才能持續交付產品及服務. 只有災難重建計畫(DRP), 是不足以回復業務或組織層級的運作的. |   在ISO 22301不用災難這詞，而是定義為中斷事故  小問題稱“事故”，導致業務中斷的稱“中斷事故”  Crisis 危機：影響到公司/個人的聲譽  危機處理通常需要公關  Emergency 緊急狀況：影響到人身安全的情況  必須優先、第一時間進行回應  老師的書 P240有說明  IM事故管理：使用一套有系統的方法，降低事故造成的負面影響   |  | | --- | | 程序要清楚 ： 前中後做哪些事？每件事要怎麼做？  組織團隊(R&R)  法律 :處理事故發生時的法律議題  資安：出事前-預防、偵測 => SOC 安全營運中心  出事時-先進行真假的判斷(資安分析師)，若為真，就要判斷優先級  判斷出優先級就要進行通報(通報的程序要寫清楚)、找資源(人力)處理  進行回應：先控制損害、再清除事故、若有損害再復原  出事後-找出根因預防再發生  IT：處理事故發生受影響的資訊系統  人資：處理事故發生時人力資源分配  公關：處理事故發生時公司或個人聲譽 |   NIST IR   |  |  |  |  | | --- | --- | --- | --- | | 準備 | 偵測與分析 | 抑制、根除與還原 | 事後行動 | | 團隊(CIRT、CIRST)  SOC  程序  計畫 | .判斷真假  .認優先級  .通報程序  .找人處理  .收集證據 | 抑制：中毒拔除網路線  根除：掃毒與解毒  還原：中毒掛了系統只能重建  收集證據 | 找到根因防止再發生  (只要重複發生就是沒有治本) | | 事故的發生可能涉及違法行為，因此需要收集證據  收集證據 :  收集的人要合格 Qualified people  要有證據保管鏈 Chain-of-custody | | | |   E-Discovery：  目的是讓公司之間可以彼此調閱數位證據  企業要保管好資料可供調閱，萬一出事時才能提出證據  E-Discovery屬於治理的一部分  公司沒有做到E-Discovery，是沒有制度，違背了due diligence     |  | | --- | | Information Governance 信息治理:確保有資訊系統有妥善的資料處理流程  Identification 識別: 當有需求被要求提供時,能指出信息發現的位置  Preservation 保存 : 確保發現的信息不能被更改或刪除 Collection 收集: 用於**E-Discovery**的過程  Processing 處理:過濾收集訊息,進行初步修剪**Cut through**,減少資訊量的檢查  Review 檢視:剩下的訊息檢查那些是移除敏感需要保護的資訊 Analysis 分析:針對留下的內容與文檔再深層的檢測 Production 產生:需提交信息產出的標準化格式  Presentation 呈現 :向證人與法院和其他當事者演示信息 |   調查(investigation): 為深入了解特定人,事,物等事實, 所採行之正式而系統化的探詢, 查驗及研究  證據(evidence): 可以支持或證明某一事件、行動或判定事實真假之資訊.  鑑識(forensics): 為法律檢調之目的所採行之證據採集, 保存及分析等作為.  老師補充   |  | | --- | | 北卡羅萊納科學與數理學院的證據的分類:  1. 實體(physical)/非實體: 子彈/血跡 vs 證詞  2. 實物(real)/展示(demonstrative): 子彈 vs 彈道圖 (實物證據跟實體證據常重疊)  3. 已知/未知: 不須實驗室檢驗 vs 須要實驗室檢驗  4. 個別/群體: DNA vs 腳印  PS. 實體的概念比實物大. 實體為實際的物理存在, 實物指可直接呈堂的東西, 因此多半也是實體證據.  OSG的分類:  1. 實物證據  2. 書面證據: 最佳(原始)證據原則及口頭證據原則(parol evidence rule). 口頭約定也是合約的一部份, 但一旦簽了書面的合約, 就不能再用口頭約定來推翻書面合約.  3. 言詞證據: testimonial evidence. 傳聞證據(hearsay)及專家證詞(expert testmony/opinion)  4. 展示證據: 用來輔助說明的設計圖, 模型, 照片, 圖表. 通常是(專家)證人在輔助證詞時搭配使用.  https://youtu.be/pW4XQM-iQWQ  直接證據(direct evidence)指在其為真實的情況下，無需經過推理或推定即可直接證明待證事實成立的證據，尤指證人就其親自經歷或感知的事實所提供的證言。與情況證據〔circumstantial evidence〕相對。  情況證據〔circumstantial evidence〕也稱為間接證據(indirect evidence), 指基於常識可以合理地從中推斷出待證事實的情況或事實，而並非個人親身經歷或親眼所見的事實。也指除證人證言以外的其他形式的證據。在無直接證據的情況下，與待證事實有關聯性的間接證據可以被採納。 ( 撰)  http://lawyer.get.com.tw/dic/DictionaryDetail.aspx?iDT=47602  http://lawyer.get.com.tw/dic/DictionaryDetail.aspx?iDT=41649 | |

二、軟體工程

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 工程強調：架構、生命週期  系統架構：元素與元素間的互動  生命週期： 階段、每個階段要做什麼  系統工程生命週期  NIST-800-64、NIST-800-160 V1、ISO15288  軟體工程生命週期  其實沒有標準定義，但是有大眾化的做法   |  |  |  | | --- | --- | --- | | **規劃：**  .確認工程方法(敏捷、反覆、瀑布)  .確認IPT(整合產品團隊)   |  | | --- | | 業務 ： 最了解需求，提出需求  開發 ： 實現需求  測試 ： 確認功能滿足需求  Ops ： 進行維運  當有這些團隊並且使用自動化工具協同作業，就是DevOps  DevOps是一種團隊協同作業文化，但要實現這種文化，需要大量的自動化工具  使用自動化工具最大的問題是，由於是自動化，在上線前很可能沒有經過授權  DevSecOps : 自動化上線之前必有安全的授權  解決自動化的風險問題 |   **需求與分析階段：**  .需求收集與引出-訪談、會議  .分析需求，整理出需求清單  .檢查需求清單  Verification-檢查正確性，內部自檢  .客戶進行確認簽字(確認需求規格)  Validation-驗證有效性，外部客戶檢查  **設計：**  .設計的第一步是設計架構  .紙上談兵的解決方案  .自行檢查設計圖(Verification)  .客戶確認設計圖(Validation)- 確認設計規格  **開發：**  .按圖施工  **測試：**  .確認功能  .確認有效  **上線與維護：**  .上線要取得授權  .持續維護 |  |     軟體工程方法   |  |  | | --- | --- | | 軟體開發生命週期不斷反覆，每次週期產生可上線的成果，就是**敏捷 (週期反覆＋價值漸增)**  (反覆的標的物是軟體開發的生命週期)  scrum稱interaction為sprint  scrum是一種agile方法  軟體開發生命週期不斷反覆，每次週期產出但不能上線的成果，就是**反覆(螺旋)**  軟體開發生命週期只跑一次，就是**瀑布**  老師案例分析與思考：   |  | | --- | | Ｑ一家新的純網銀正在開發其核心系統，以支持網銀業務。業務人員認為系統應在經過全面測試後再提交驗證(certification)，而IT團隊則堅持應先發布系統以進行評估，再根據反饋進行改進。 但是，IT部門的方法導致CEO辭職，因為核心系統無法滿足最低的安全要求，並且始終無法從監管機構獲得運營許可。當初應該採用以下哪種開發方法來避免這樣的失誤？  A. Waterfall  B. Incremental development  C. Agile  D. Continuous delivery  說明：  敏捷是一種思維，它包括一組價值觀、原則和實踐。  它強調提供有效的軟件/價值（又名增量）、人員協作和風險適應。  增量開發（頻繁發布以創造價值）是敏捷的一部分。  持續交付是敏捷中的一種普遍做法，意味著“自動”發布和“頻繁”發布。    綜上所述，問題中的IT部門在某種程度上遵循了敏捷方法（包括增量開發和持續交付），但這不起作用，並且導致CEO辭職。  敏捷不是萬靈丹， 瀑布也不是邪惡的代名詞。  它仍然被廣泛採用並有效地解決了問題。  理解“迭代”和“增量”的基本概念是解決此問題的關鍵 | |   DevOps   |  | | --- | | 個人：  品質保證(測試)、營運、開發的交集  強調 溝通、合作、自動化、變更管理、組態管理、減少錯誤  目標是目的在有效的交付產品，滿足客戶需求，達成業務目標  風險是自動話可能越過授權直接上線  Joy補充  主要是將營運、開發、測試之間的間隔去除，在透過自動化設定、測試，讓開發團隊的程式可透過自動化進行測試及部署，在於營運團隊，也不需反覆與開發人員確認設定文件，避免造成部署錯誤。  三者合作，可增快交付速度，中間過程也能確保交付的程式，是有經過測試，部署時，也是設定正確。  老師補充  .DevOps其實是口水字. 以前開發完成, 上線後由IT接手維運. 過程沒有好的自動化工具支援. 後來有了這些工具, 從開發到部署上線都有自動化工具支援, 所以人工減少, 效率提升, 人為錯誤也降低了. 後來這些工具廠商就創了DevOps這個東西來行銷產品.  .軟體不外乎, 開發人員(Dev), 測試人員(Testing)及維運人員(Ops)  .Dev一般把開發跟測試含在裏面  .後來又在炒作, DevOps不能忽視資安,所以要Dev+Sec+Ops, 又有了DevSecOps  一切都是口水  .DevOps的"標準"說法: 因為DevOps將開發及維運團隊納入, 需要透過工具及科技的促進協同作業及溝通, 而因為與組織不同的人有關, 所以需要搭配文化的變革, DevOps才能發揮最大的效益. |   軟體架構   |  |  | | --- | --- | | UI(呈現層)  運算(業務邏輯層)  資料(資料存取層) | API、UI  服務  SQL | |  | |   網站用SOA (服務導向架構)  <http://www.cc.ntu.edu.tw/chinese/epaper/20070620_1008.htm>  軟體架構發展趨勢   |  | | --- | |  |   關連式資料庫 RDBMS   |  |  | | --- | --- | | .關聯是學術名詞，不是關係，強調的是表格  .表格式資料庫  .豆腐狀的表格，不能合併儲存格  .每個表格都要有資料，一個表格一個資料  .表格當中的主鍵，不能重複  .**主鍵不重複，確保實體不重複，又叫實體完整性，主鍵的目的是為實體完整性**  .表格豆腐狀、資料一個蘿菠一個坑、有唯一識別的主鍵，叫做第一正規化 | 一張含有 螢幕擷取畫面 的圖片  自動產生的描述 | | 關係則是表格之間的連結  一張含有 螢幕擷取畫面 的圖片  自動產生的描述  1對多關係 : 主表格的一筆資料，會對應明細表格的多筆資料 | | | 語意完整性：表格裡的資料必須對人有意義，在技術上控制資料型態、資料範圍、資料長度  孤兒紀錄：明細表格的資料找不到主表格  參考完整性：明細表格的連外鍵必需對應到主表格的主鍵 | | |

三、雲運算

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 老師的補充   |  | | --- | | 1. 有很多電腦提供運算, 不一定能滿足雲的定義. ISO跟NIST都有準確的"雲運算"定義.   2. "雲服務"是指將這些運算能力包裝成服務的形式, 也就是"服務模式"(service model).  如IaaS, PaaS, SaaS等  IaaS:提供VM、VM相關元件、單一空間  SaaS:提供應用軟體，直接由End user使用，  PaaS:提供應用程式運行的平台(PHP、ASP)，過程必有程式碼的上傳  3. 這些"雲服務"要提供給誰用, 就是所謂的"部署模式"(deployment model).  如public cloud, private cloud, community cloud, hybrid cloud等. |   NITS 800-145 cloud  <https://csrc.nist.gov/publications/detail/sp/800-145/final>   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | * 雲端的定義   自助式服務 (On demand self-service)  隨時網路存取 (Broad network access)  想用資源就用、釋放與收回 (Resource pooling)  快速有彈性、可伸縮 (Rapid Elasticity)  可計價/衡量的服務 (Measure service)  ISO17888多一個定義：多住戶 (Multi tenancy)   * 服務模式 (服務提供方式)   软件即服务（SaaS）。   |  | | --- | | 提供消費者在雲基礎架構上運作的應用程式  可透過瀏覽器或應用程式介面使用應用程式  使用者能管理應用程式的基本設定  應用實例：Office365、Adobe CC、Gmail、Google Drive、OneDrive |   平台即服务（PaaS）。   |  | | --- | | 提供消費者在雲基礎架構上運作的程式(PHP、ASP)服務平台(Mysql＋Apache、IIS+MSSQL)  可將程式碼部署到服務平台運行  使用者能管理服務平台的基本設定(平台環境、權限)，但不一定能進行平台更新  應用實例：Google APP Engine |   基础架构即服务（IaaS）。   |  | | --- | | 提供消費者在雲基礎架構上的處理、儲存、網路等基本運算資源  可以選擇安裝需要的作業系統、服務平台、應用程式  使用者能完整管理作業系統與安裝在作業系統上的所有平台與應用程式，並能有限度的控制網路(虛擬switch、虛擬防火牆)  應用實例：AWS S3、AWS EC2、Google GCP |  * 部署模式 (提供給誰使用)   Public Cloud :   |  | | --- | | (1)提供給大眾使用  (2)基礎架構、平台由供應商所有 |   Private Cloud :   |  | | --- | | (1)提供給企業自用  (2)基礎架構、平台由企業或是供應商所有 |   Hybrid Cloud :   |  | | --- | | (1)同時提供給大眾使用與企業自用  (2)由兩種以上的部屬模式組成  (3)基礎架構、平台，由企業或是供應商所有 |   Community Cloud ：   |  | | --- | | (1)提供給一個有共同目標、共同利益的集團、聯盟、政府不同行政部門  (2)基礎架構、平台由集團、聯盟、政府不同行政部門或供應商所有 | |   雲的共同責任模型  IaaS: 作業系統以上由客戶負責  PaaS: 應用程式(含資料庫設定)以上由客戶負責  SaaS: 資料由客戶負責  一張含有 計算機, 標誌, 坐, 街道 的圖片  自動產生的描述  老師補充  <https://wentzwu.com/2020/08/30/privacy-and-cloud-services/> |

**Cloud access security brokers (CASBs)**

|  |
| --- |
| **Cloud access security brokers (CASBs)** are on-premises, or cloud-based security policy enforcement points, placed between cloud service consumers and cloud service providers to combine and interject enterprise security policies as the cloud-based resources are accessed.  CASBs consolidate multiple types of security policy enforcement. Example security policies include authentication, single sign-on, authorization, credential mapping, device profiling, encryption, tokenization, logging, alerting, malware detection/prevention and so on.  云访问安全代理（CASB）是内部部署或基于云的安全策略实施点，位于云服务使用者和云服务提供商之间，以在访问基于云的资源时合并和插入企业安全策略。  CASB整合了多种类型的安全策略实施。 示例安全策略包括身份验证，单点登录，授权，凭据映射，设备配置文件，加密，令牌化，日志记录，警报，恶意软件检测/预防等。  一張含有 螢幕擷取畫面 的圖片  自動產生的描述  個人解釋  CASB  是一個安全服務代理人  是介於雲端服務與用戶之間的安全服務介面  提供安全政策、狀態檢測、資料安全..安全服務 |

四、無線網路

|  |
| --- |
| Ad-hoc mode : 設備對設備  Infrastructure mode : 很多設備對一個基地台    wifi攻擊  War Driving : 開車到處收集各無線基地臺的弱點  War Chalking : 選定目標後進行弱點標記  Evil Twin : 使用惡意基地台模仿一個真的SSID  Bluetooth  Bluejacking 攻擊者可以發送惡意訊息或是圖片給受害者  Bluebugging 攻擊者可以透過藍芽對受害者手機執行命令，如打電話、發簡訊、修改資料...  Bluesnarfing 攻擊者可以透過藍芽存取受害者行事曆、照片、影片.. |

資料庫專有名詞

|  |
| --- |
| Database  資料的集合  有一定格式、組織、系統的資料紀錄檔  Database schema  定義資料的儲存結構和那些資料需要建立索引  Data dictionary  ( Metadata ) 描述資料的資料 |

資料庫安全議題

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 安全議題  Aggression 聚合  -透過較低安全級別的資料組合來得出較高安全級別(原本沒權限存取)的資料  -通常藉由軟體工具、SQL的組合資料函數  -解法是嚴格控制SQL聚合函數的使用權  Inference 推理  -透過一些已知非敏感的訊息，推測出原本無權知道的敏感訊息  -有高度信心根據”已知”的合法資料推測出需要授權才能知道的事  -通常藉由人的推理能力  老師補充  別把Aggregation跟Inference這個二看得太複雜.  Aggregation聚合其實就只是收集資料而已, 就像拼圖一樣, 當你拼到一定程度, 不用想, 一眼就可看出更高等級的機密資料就是聚合.  如果還要想一想, 猜一猜, 就叫推理.  反正就是收集資料, 不用想的就聚合, 還要猜的叫推理.  Content dependent、Context dependent  依照內容或是前後關係來進行判斷  Data warehousing  將不同來源的資料加以儲存，並產生 Metadata  包含大量易受到聚合、推論攻擊的潛在敏感信息  Data mining  分析人員使用工具對Data warehousing的資料加以分析，尋找關聯關係或是趨勢  搜索安全相關的數據，找出正在進行的攻擊、損害、破壞等異常事件  安全管理   |  |  | | --- | --- | | Cell suppression | 將資料隱藏避免推理攻擊  隱藏特定欄位 | | Partitioning | 將資料分割避免未經授權者存取所有資訊 | | Noise and perturbation 擾動 | 插入假資料來迷惑 / 干擾攻擊者  有權限看到真資料，無權限看到假資料 | | Database view | 不同群組/角色看到"應該"看到的資料  Need to Know | | Content dependent access control | 依資料機密性來判斷，越機密越少人知道 | | Context dependent access control | 紀錄並判斷存取動作是否適切 | | Polyinstantiation 多實例 (有真有假) | *一種資料保護技術. 將一個實體相關的資料, 複製多份, 並變更複本的內容給不同的對象看, 以對原始資料進行保密.*  避免推理攻擊  防止低權限的人察覺高權限才可以知道的事 |  |  | | --- | | *Source: NIST SP 800-8 (obsoleted)*  ***Inference****: Derivation of new information from known information.*  *The inference problem refers to the fact that the derived information may be classified at a level for which the user is not cleared.*  *The inference problem is that of users deducing unauthorized information from the legitimate information they acquire.*  ***Inference***推论：从已知信息中派生新信息。  推论问题是指这样的事实，即可以在不清除用户的级别上对导出的信息进行分类。  推论问题是用户从他们获取的合法信息中推断出未经授权的信息。  ***Aggregation****: The result of assembling or combining distinct units of data when handling sensitive information. Aggregation of data at one sensitivity level may result in the total data being designated at a higher sensitivity level.*  ***Aggregation***汇总：处理敏感信息时组合或合并不同数据单元的结果。  以一个敏感度级别聚合数据可能会导致以较高敏感度级别指定总数据。  *Polyinstantiation: Polyinstantiation allows a relation to contain multiple rows with the same primary key; the multiple instances are distinguished by their security levels.*  多重实例化：多重实例化允许一个关系包含多个具有相同主键的行；多个实例通过其安全级别进行区分。  *老師補充:*  *一種資料保護技術. 將一個實體相關的資料, 複製多份, 並變更複本的內容給不同的對象看, 以對原始資料進行保密.*  *Bruce Sybex OSG有舉一個軍方的例子. 美國核潛艦要去北極海出極機密的任務, 任務資料不能對後勤人員隱藏, 不然反而會引起大家注意, 知道這次是極機密的任務. 美國軍方的作法是另外生成一筆任務資料(假的), 讓一般權限的相關人員看, 而不揭露真正的極機密資料.*  *18:29 Bruce 多實例, 就是一個實體(有ID的東西, 如任務)有多個副本.  Referential integrity: A database has referential integrity if all foreign keys reference existing primary keys.* 参照完整性：如果所有外键都引用现有的主键，则数据库具有参照完整性。  *Entity integrity: A tuple in a relation cannot have a null value for any of the primary key attributes.* 实体完整性：关系中的元组不能对任何主键属性具有null值。  *Granularity: The degree to which access to objects can be restricted. Granularity can be applied to both the actions allowable on objects, as well as to the users allowed to perform those actions on the object.*  粒度：可以限制访问对象的程度。粒度既可以应用于对象上允许的动作，也可以应用于被允许对对象执行那些动作的用户 | |

老師補充 IPS/IDS

|  |
| --- |
| 一個可靠的IDS要具備敏感度(sensitivity)及準確度(specificity)這二個能力:  1. 有足夠的敏感度去識別異常流量，不會亂發警報  2. 有足夠的準確度去織別正常流量，不會錯放駭客  IDS依偵測能力可分為signature-based及anomaly-based二種。異常(anomaly)又稱為偏離值(outliner), 是統計學的用語。例如，一個統計的常態分配(鐘型)包含了全部的樣本(即100％)，但愈往左右二側出現的樣本數愈少。我們很常把中間區域的95％視為正常，而超出正常區域的5%視為異常，即anomaly或outliner. (左、右各2.5％, 共5％).  但出現在左右偏離區域的異常值不一定真的是異常。例如平常心跳每分鐘120下，超出一般人所認知的正常值很多，這通常會被視為異常，甚至被視為有心血管疾病。然而，有極少數的人有特殊體質，這種心跳率在他身上是正常，不是疾病。遇到這種特殊體質，就會有誤判的情況。相反的，即使落在正常區域，也會有誤判的情況。  收集一定的資料或樣本，我們就可以畫出一個統計的分配/布(建立/訓練模型), 然後用這個分配/模型來預測/分類是否為正常/異常。一個好的模型，錯誤率較低，正確率較高; 也就是有較高的敏感度(高正確示警率)及準確度(高正確放行率)。  為了檢驗一個模型的好壞，我們會把判斷的結果收集起來，再畫成統計的分配，通常是去討論示警及放行的案例中，有多少是正確/錯誤示警，有多少是正常/錯誤放行。敏感度(sensitivity)及準確度(specificity)就是正確示警及正確放行的指標。以下維基百科的文章有更多的模型評估指標:  <https://en.m.wikipedia.org/wiki/Sensitivity_and_specificity>  ROC曲線則是專門用在評估二元分類模型(正常信/不是正常信，真酒/不是真酒，我機/不是我機)。  <https://en.m.wikipedia.org/wiki/Receiver_operating_characteristic> |