安全性軟體開發流程概論

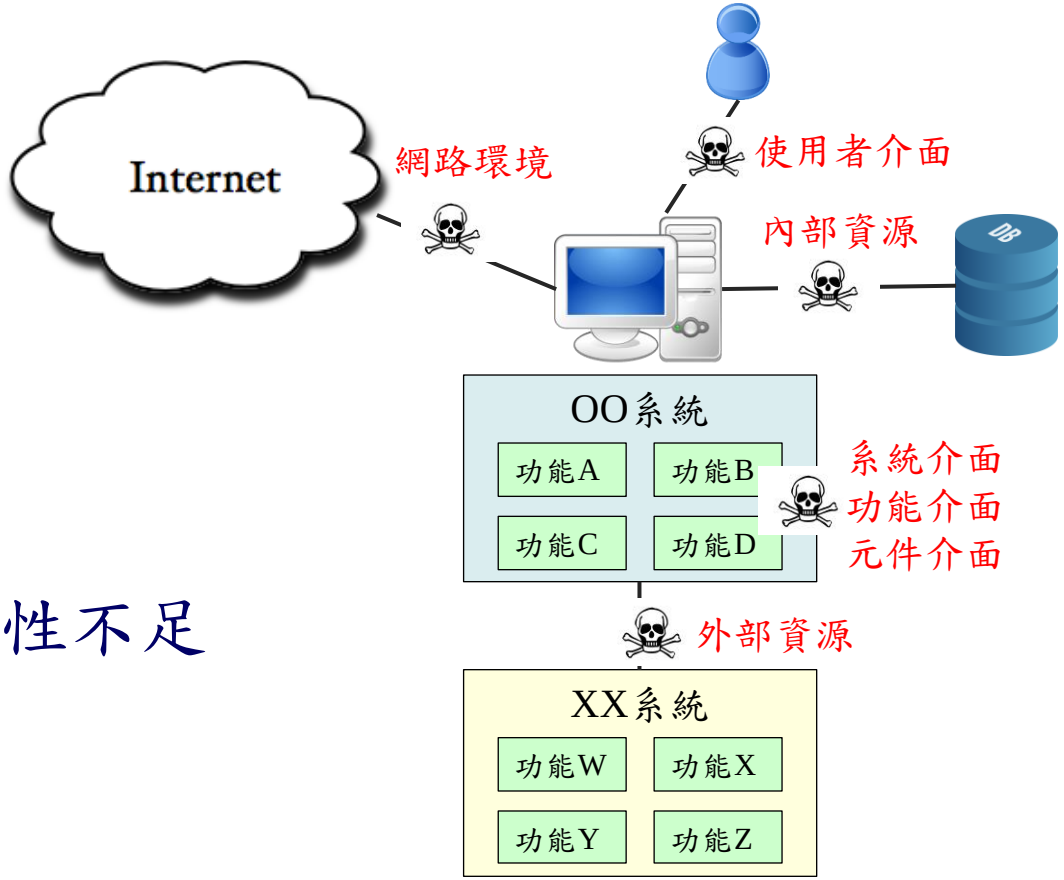
(*Secure Software Development Life Cycle Concept*)

報告人：王耀華

中華民國106年8月1日

軟體安全之重要性

- **資安事件居高不下**
  - 網路入侵攻擊
  - 軟體本身的漏洞
- **軟體系統威脅**
  - 軟體系統特性改變
  - 軟體系統環境安全性不足



--> **軟體安全需全面性考量**

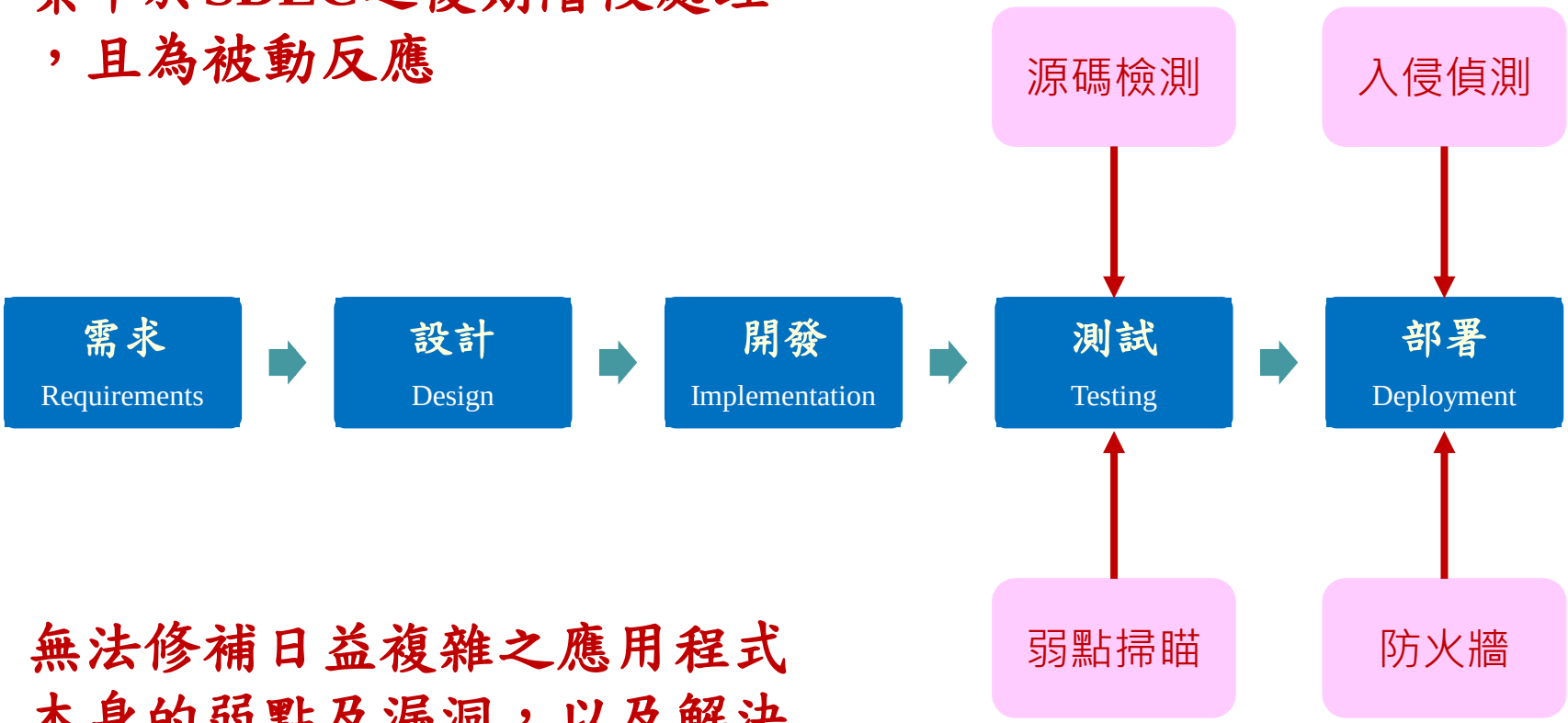
--> **需有一套完整可行的軟體安全方法論**

SDLC (Software Development Life Cycle)

需求 Requirements	設計 Design	開發 Implementation	測試 Testing	部署 Deployment
需求訪談、需求分析、研擬可行性方案與排定優先順序	軟體架構、邏輯設計及細部設計	程式開發	驗證程式開發結果符合需求	系統上線與後續維護
• 需求訪談會議 • 建置 Prototype • RFP • Use Case …	• 結構化分析設計 • 物件導向分析設計 …	• 函式 • 程序 • 模組 • 物件 …	• 單元測試 • 整合測試 • 系統測試 • 驗收測試 • 功能測試 • 效能測試 • 壓力測試 …	• 硬體安裝 • 系統安裝 • 環境設置 • 教育訓練 • 問題分析與處理 • 變更管理 • 系統轉移 …

SDLC面對安全性的補救措施

集中於SDLC之後期階段處理，且為被動反應



無法修補日益複雜之應用程式本身的弱點及漏洞，以及解決系統設計上的問題

安全軟體開發生命週期 (SSDLC)

安全軟體需求	安全軟體設計	安全軟體開發	安全軟體測試	安全軟體部署
需求萃取	遵循安全設計原則並進行威脅建模與風險評估	避免常見軟體漏洞並發展因應控制措施	驗證安全性需求並確保達成安全目標	確保安全控管機制與防護措施正常運作
RFP	威脅建模	SANS、WASC	源碼檢測	組態管理
Prototyping	威脅風險評估	OWASP Top 10	弱點掃瞄	變更管理
資料分級	發展控制措施	Vulnerability Database	安全需求驗證	持續整合
OWASP ASVS	安全設計審查	Secure Coding Style		
需求追溯表	安全架構分析	Secure Code Review		

軟體安全特性

機密性	完整性	可用性
身分認證	授權	稽核
會話管理	錯誤及例外處理	組態管理

CSTI 資安科技研究所 CyberSecurity Technology Institute

軟體安全特性

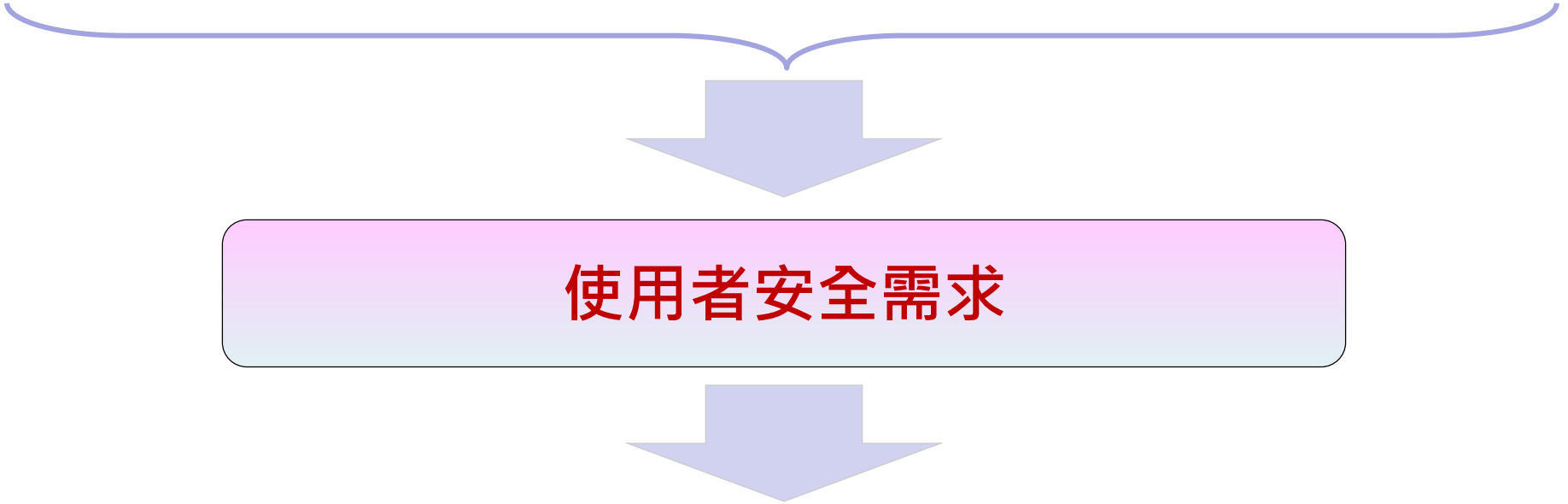
確保只有經授權的人才可取得資訊，避免資訊洩露	確保資訊不受未經授權的竄改與資訊處理方法的正確性	確保經合法授權的使用者，在需要時可以取得資訊，並使用相關資產
機密性 Confidentiality	**完整性 Integrity**	**可用性 Availability**
確認使用者的身分識別確實為其所宣告的的主體	確認已認證的實體被允許或禁止對特定資源進行特定動作的程序	協助建構出使用者行為的歷史紀錄，並滿足可用性標準(具不可否認性)
身分認證 Authentication	**授權 Authorization**	**稽核 Auditing**
避免會話洩露、劫奪或竄改，確保會談過程行為受到保障	明確說明處理錯誤及例外的要求，以避免資訊洩漏威脅	建立或維持軟體功能、效能及相關特性的一致性和穩定
會話管理 Session Management	**錯誤及例外處理 Error and Exception Handling**	**組態管理 Configuration Management**

CSTI 資安科技研究所 CyberSecurity Technology Institute

安全軟體需求

萃取安全需求

軟體安全政策 組織安全目標	Prototyping	需求建議書 RFP	常見威脅	誤用案例
資料分級 安全需求查檢表	ASVS	設計階段 活動的回饋	常見弱點	使用案例



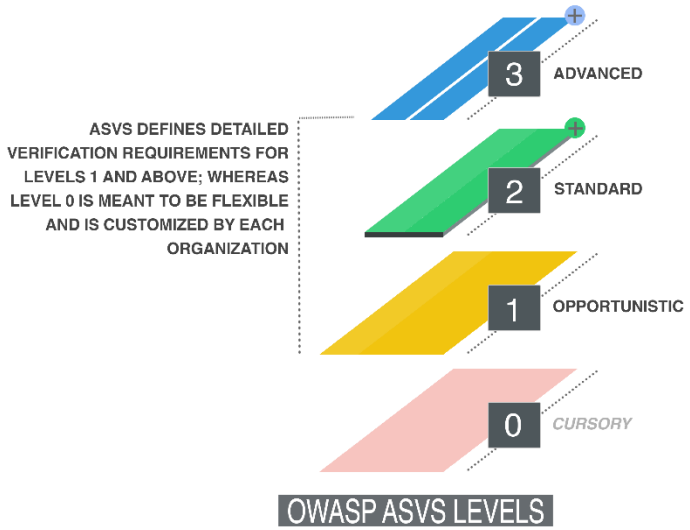
使用者安全需求

需求追溯表 (RTM)
Requirements Traceability Matrix

CSTI 資安科技研究所
CyberSecurity Technology Institute

OWASP – ASVS

- **ASVS** (Application Security Verification Standard)
- 透過 ASVS 可以提供 Web 應用程式相關人員一個共通性的語言，清楚的標示出必須完成哪些安全性的需求。
- ASVS 將所有需要檢驗的項目分成 3 個層級，層級越高表示檢驗項目的數量與深度也隨之增加。



適用於關鍵性應用軟體，需要大量安全驗證程序

適用於處理重要的企業對企業交易的應用程序，實施關鍵業務或敏感功能的應用程序。

基本一定要符合的，適用所有網路存取軟體

資料來源： OWASP Application Security Verification Standard 3.0.1

OWASP – ASVS 3.0.1

V1. Architecture, design and threat modelling 架構、設計與威脅建模
V2. Authentication 授權
V3. Session management 會話管理
V4. Access control 存取控制
V5. Malicious input handling 惡意輸入處理
V6. Output encoding / escaping (已於2.0時併入V5)
V7. Cryptography at rest 儲存階段密碼學應用
V8. Error handling and logging 錯誤管理及紀錄
V9. Data protection 資料保護
V10. Communications 通訊
V11. HTTP security configuration HTTP 安全組態
V12. Security configuration verification requirements (已於2.0時併入V11)
V13. Malicious controls 惡意控制
V14. Internal security verification requirements (已於2.0時併入V13)
V15. Business logic 商業邏輯
V16. File and resources 檔案與資源
V17. Mobile 行動裝置
V18. Web services (NEW for 3.0)
V19. Configuration (NEW for 3.0)

可用來定義開發需求，也可用來規劃資訊安全測試的範圍、測試個案及安全性驗證

資料來源： OWASP Application Security Verification Standard 3.0.1

CSTI 資安科技研究所 CyberSecurity Technology Institute

V1: Architecture, design and threat modelling (1/2)

#	Description	1	2	3	Since
1.1	Verify that all application components are identified and are known to be needed.	✓	✓	✓	1.0
1.2	Verify that all components, such as libraries, modules, and external systems, that are not part of the application but that the application relies on to operate are identified.		✓	✓	1.0
1.3	Verify that a high-level architecture for the application has been defined.		✓	✓	1.0
1.4	Verify that all application components are defined in terms of the business functions and/or security functions they provide.			✓	1.0
1.5	Verify that all components that are not part of the application but that the application relies on to operate are defined in terms of the functions, and/or security functions, they provide.			✓	1.0
1.6	Verify that a threat model for the target application has been produced and covers off risks associated with Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of privilege (STRIDE).			✓	1.0
1.7	Verify all security controls (including libraries that call external security services) have a centralized implementation.		✓	✓	3.0
1.8	Verify that components are segregated from each other via a defined security control, such as network segmentation, firewall rules, or cloud based security groups.		✓	✓	3.0
1.9	Verify the application has a clear separation between the data layer, controller layer and the display layer, such that security decisions can be enforced on trusted systems.		✓	✓	3.0
1.10	Verify that there is no sensitive business logic, secret keys or other proprietary information in client side code.		✓	✓	3.0
1.11	Verify that all application components, libraries, modules, frameworks, platform, and operating systems are free from known vulnerabilities.		✓	✓	3.0.1

資料來源： OWASP Application Security Verification Standard 3.0.1, July 2016

ASVS與軟體安全特性

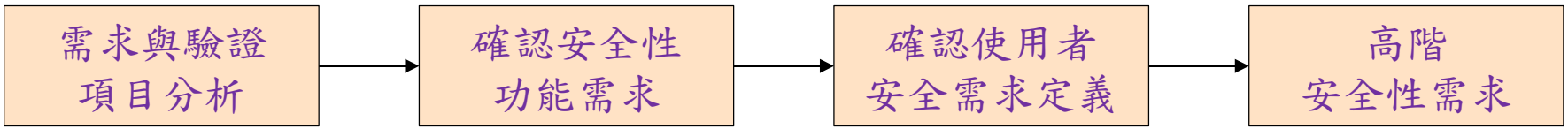
ASVS 3.0.1	機密性 Confidentiality	完整性 Integrity	可用性 Availability	身分認證 Authentication	授權 Authorization	稽核 Auditing	會話管理 Session Management	錯誤及例外處理 Error and Exception Handling	組態管理 Configuration Management
V1. Architecture, design and threat modelling 架構, 設計與威脅建模	○	○	○	○	○	○	○	○	○
V2. Authentication 身分認證	○			○				○	○
V3. Session management 會談管理							○		
V4. Access control 存取控制					○	○		○	
V5. Malicious input handling 惡意的輸入控制								○	
V7. Cryptography at rest 儲存階段密碼學應用	○							○	
V8. Error handling and logging 錯誤處理與與紀錄	○	○			○	○		○	
V9. Data protection 資料保護	○	○	○			○			
V10. Communications 通訊安全	○	○		○	○				
V11. HTTP security configuration HTTP 協定安全組態	○	○		○					
V13. Malicious controls 惡意控制	○	○	○	○	○	○	○	○	○
V15. Business logic 商業邏輯	○	○	○	○	○	○	○	○	○
V16. File and resources 檔案與資源	○	○	○		○				
V17. Mobile 行動裝置	○				○				
V18. Web services Web 服務	○	○		○	○		○		
V19. Configuration 組態	○	○		○	○				○

CSTI 資安科技研究所 CyberSecurity Technology Institute

案例說明

- 本案例為公司內部使用之活動查詢網站，使用者通過登入驗證功能，可查詢公司內部之活動資訊列表
- 功能項目分析如下表說明：

功能編號	功能項目	需求功能描述
SRS-A-01	登入驗證	匿名使用者透過登入驗證功能，可成為登入使用者
SRS-B-01	活動列表	登入使用者可查看活動列表



實作案例：需求追溯

編號	需求	設計	實作
SR-05	所有輸入應進行驗證，輸出前應編碼	使用白名單方式進行輸入驗證，並使用正規表達式驗證所輸入的資料 (SRD-05-01)	白名單與正規表達式驗證方式需實作於輸入驗證模組 (SRI-05-01-001)
			登入驗證功能需使用輸入驗證模組進行處理 (SRI-05-01-002)
			活動列表功能需使用輸入驗證模組進行處理 (SRI-05-01-002)
		輸入資料需防範SQL Injection攻擊 (SRD-05-02)	登入驗證功能需使用輸入驗證模組及參數化的預存程序進行資料庫存取 (SRI-05-02-001)
		使用 HTMLEncode 和 URLEncode 函數對輸出進行編碼 (SRD-05-03)	HTMLEncode 和 URLEncode 需實作於輸出驗證模組 (SRI-05-03-001)
			活動列表功能需使用輸出驗證模組進行處理 (SRI-05-03-002)

Requirement (by user language description) → **Design** (by designer description) → **Implementation** (by programmer description)

CSTI 資安科技研究所 CyberSecurity Technology Institute

安全軟體設計

安全軟體設計作業程序

安全軟體設計原則

最弱環節 Weakest Link	簡化設計 Economy of Mechanism	使用現存元件 Leveraging Existing Components
縱深防禦 Defense in Depth	安全故障 Fail Secure	防範單點失效 Single Point of Failure
最小權限 Least Privilege	完全仲裁 Complete Mediation	最少共用機制 Least Common Mechanism
權限分離 Separation of Duties	開放設計 Open Design	可接受度 Psychological Acceptability

↓

安全架構分析

↓

資料流程圖 (EDFD)

↓

威脅建模 (STRIDE)

↓

威脅評估 (DREAD)

↓

發展降低風險之控制措施

↓

安全設計審查

CSTI 資安科技研究所 CyberSecurity Technology Institute

安全軟體設計原則 (1/2)

設計原則	說明
最弱環節 Weakest Link	軟體設計時應從攻擊者的角度考量目前軟體架構或流程中，防護最不足甚至防護可能被繞過的情況，進行控制措施補強。 可透過源碼檢測、主機弱點掃瞄、網站弱點掃描及滲透測試等方式，找出弱點或防護不足之處，以進行安全控制措施強化工作。
縱深防禦 Defense in Depth	單一的安控措施被突破或失效不會造成安全危害或只造成部分的危害，在軟體單一環節或進入路徑採用多重的安控措施，又稱作分層防禦 (Layered Defense)。 – 為避免帳號密碼容易被竊取，採用多重因素認證，例如：增加使用 OTP (One Time Password)機制、帳號密碼＋圖形驗證、軟體憑證登入＋圖形驗證…等。 – 實體主機則可採用 IDS + IPS + Firewall 來增加防禦縱深。
最小權限 Least Privilege	任何使用者或程序僅在被允許的最短時間區間內，被給與必需的、最小的存取權限，以完成其作業任務。 – 如僅需查詢資料，則只提供讀取權限即可，不需多給予編輯權限。 – 可透過存取控制表(Access Control List)來檢視權限設定是否適當。
權限分離 Separation of Duties	設計軟體功能時將條件設定在兩個或多個以上，且需要滿足所有條件才能完成作業。 使用某一項功能，除需先行使用帳號密碼登入系統外，另需先行提出申請，經管理者簽核同意後，才可使用該項功能。
簡化設計 Economy of Mechanism	軟體設計越複雜，越容易產生安全漏洞，在安全分析中也更不容易被找出問題。 模組化的設計、可重複使用的元件與服務、減少不必要的功能與服務。
安全故障 Fail Secure	軟體發生錯誤或故障時，預設狀態為不允許存取，且維持機密性、完整性與可用性，軟體中重要的功能或交易行為皆應設想發生錯誤的狀況，並考慮在錯誤的狀況下，各項安全機制是否還處於有效的狀態。 例如：連續登入失敗 5 次，則鎖定帳號 10 分鐘。

資料來源：行政院國家資通安全會報技術服務中心-安全軟體發展流程指引

安全軟體設計原則 (2/2)

設計原則	說明
完全仲裁 Complete Mediation	每一次主體存取物件時，都重新檢查其合法授權。 軟體中重要的物件或功能，在每次存取時，重新要求使用者的身分認證及合法授權，以避免重送攻擊(Replay Attack)的危害。
開放設計 Open Design	安全設計不應依賴「隱藏式的安全」(Obscurity) (指安全性建立於別人不知道安全作法為何)。 例如：如需加密保護資料時，應使用已公開且未被破解之加密演算法，不應使用自行設計之加密演算法。
使用現存元件 Leveraging Existing Components	重複使用現有、經過安全測試或實證的元件，得以避免增加新元件所產生的風險。 評估採用一項技術或元件時，除了其功能是否符合需求，也應考慮該項技術的安全性，以及過往是否發生安全事故。
防範單點失效 Single Point of Failure	安全設計應確保軟體不會有單點故障或錯誤造成全面停擺的可能，為達成高可用性的目的，實務上通常採取重複(Redundancy)方式進行備援處理。 可視實際需要，採取備援機制。
最少共用機制 Least Common Mechanism	盡可能減少適用於所有使用者或角色的機制，共享的機制通常代表共享的資料，必須被很小心的設計以避免資訊流破壞安全性。 – 明確區分前、後端管理介面與網址。 – 後端管理則應加強安全性身分驗證。
可接受度 Psychological Acceptability	考量安全機制對於人員的可接受度，良好的安全機制應盡可能的設計為容易使用、不影響原先的存取，以及盡可能作到讓使用者感受不到其存在，以增加轉換與接受程度。 安全機制常會影響到軟體的易用程度，使得步驟增加或是額外的動作，但應該儘量將影響最小化在合理範圍內。

資料來源：行政院國家資通安全會報技術服務中心-安全軟體發展流程指引

風險處理活動 (ISO 27005) (2/2)

處理方式	活動	實施指南
風險降低	經由被選擇的控制機制來降低風險，一直到殘餘風險被接受。	應選擇適當與合理的控制措施，以滿足風險評估和風險處置中識別的要求。
風險保留	根據風險評估，決定不採取進一步的活動而保持風險	如果風險等級符合滿足風險接受準則則不需要實施額外的控制措施。
風險避免	應該規避導致特定風險的活動或條件。	識別的風險太高，或實施其他控制處理成本超過了效益，必須改變或調整控制作為。
風險轉移	根據風險評估結果，對於特定風險最有效的管理，可能是將風險轉移給其他方。	移轉過程會牽涉組織外的夥伴，造成極有風險控制機制的改變可能會引起其他風險，因此必須考量特殊情況下的風險處置。可納入保險機制，或是與夥伴間簽訂保障合約，請求發生風險失控時的支援。

軟體設計風險分析 – 威脅建模 (微軟)

- 威脅建模採用系統化的方法，以攻擊者的角度，識別可能影響軟體系統的威脅並進行評估。基於對架構與設計的了解，識別與評估威脅後，以風險高低的順序對威脅發展適當的控制措施

識別資產 (Identifying Assets)

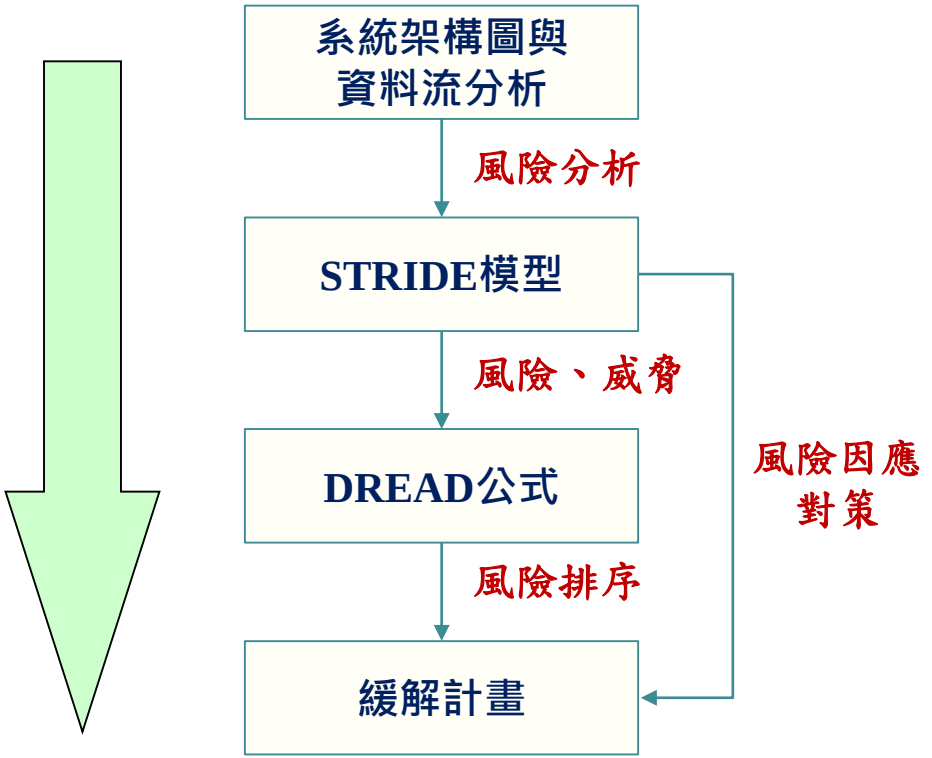
建立架構概觀 (Create an Architecure Overview)

分解應用程式 (Decompose the Application)

識別威脅 (Identifying the Threads)

記錄威脅 (Document the Threads)

評估威脅 (Rate the Threads)

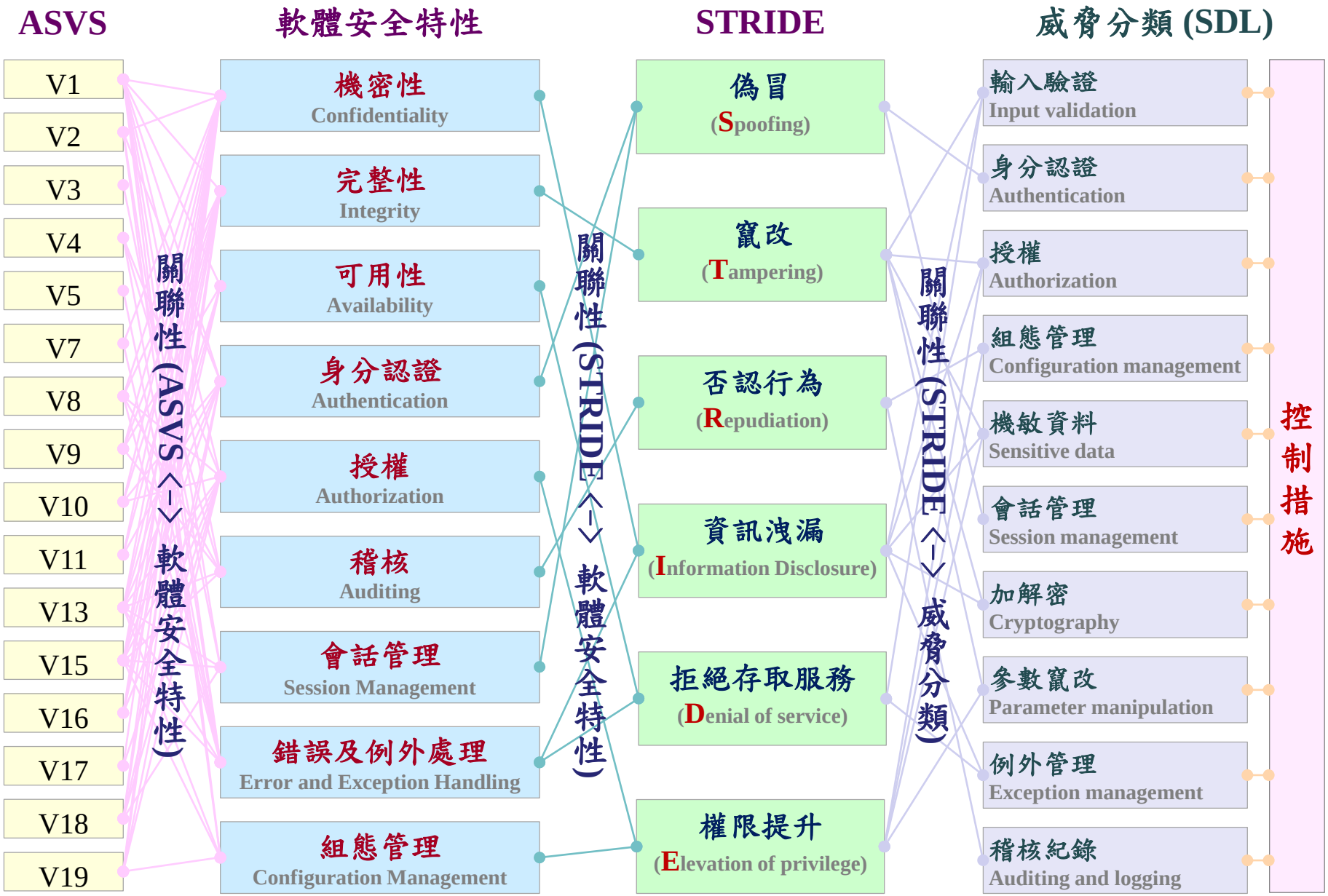


識別弱點與威脅 (STRIDE)

字首	代表含意	描述	軟體安全特性	威脅分類
S	偽冒 (Spoofing)	假冒為某人或某物	身分認證 會話管理	• 身分認證 • 會話管理
T	竄改 (Tampering)	惡意的修改資料	完整性	• 輸入驗證 • 授權 • 機敏資料 • 加解密 • 參數竄改
R	否認行為 (Repudiation)	使用者可以否認曾經進行某行為 沒有方法可以證明其行為	稽核	• 組態管理 • 稽核紀錄
I	資訊洩漏 (Information Disclosure)	資訊被洩漏給不被預期可存取的個體	機密性 錯誤及例外處理	• 輸入驗證 • 授權 • 機敏資料 • 加解密 • 例外處理
D	拒絕存取服務 (Denial of service)	對使用者拒絕服務或降低服務水準	可用性 錯誤及例外處理	• 輸入驗證 • 例外處理
E	權限提升 (Elevation of privilege)	未正常授權取得權限能力	授權 組態管理	• 授權 • 組態管理 • 機敏資料 • 參數竄改

資料來源： http://www.microsoft.com - Introduction to SDL Threat Modeling

ASVS與威脅分類之關聯 - SDL



威脅分類說明 (1/2)

威脅分類	可能威脅	
輸入驗證 (Input Validation)	緩衝區溢位	Buffer overflow
	跨站腳本攻擊	Cross-site scripting
	SQL 注入	SQL injection
	不同格式標準	canonicalization
身分驗證 (Authentication)	網路竊聽	Network eavesdropping
	暴力破解攻擊	Brute force attacks
	字典窮舉攻擊	Dictionary attacks
	Cookie 重放攻擊	Cookie replay attacks
	憑證竊取攻擊	Credential theft
授權 (Authorization)	權限提升	Elevation of privilege
	機敏資料洩漏	Disclosure of confidential data
	資料竄改	Data tampering
	引誘攻擊	Luring attacks
組態管理 (Configuration Management)	未經授權存取管理界面	Unauthorized access to administration interfaces
	未經授權存取配置檔	Unauthorized access to configuration stores
	取得控制檔內的明文機敏資料	Retrieval of plaintext configuration secrets
	缺乏個人歸責	Lack of individual accountability
	過度授權的流程和服務帳戶	Over-privileged process and service accounts

資料來源： http://www.microsoft.com - Improving Web Application Security: Threats and Countermeasures

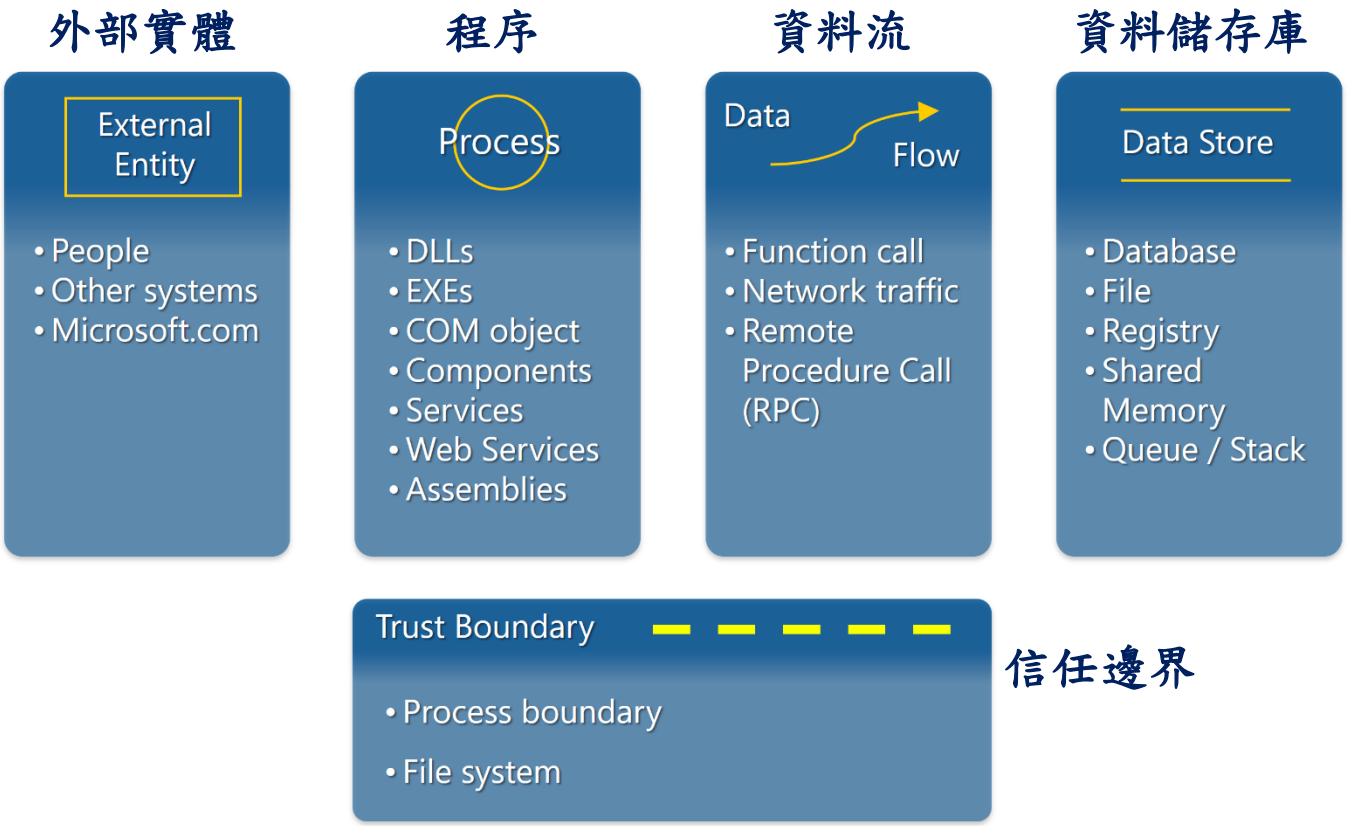
威脅分類說明 (2/2)

威脅分類	可能威脅	
機敏資料 (Sensitive Data)	存取敏感資料	Access to sensitive data in storage
	網路竊聽	Network Eavesdropping
	資料竄改	Data Tampering
會話管理 (Session Management)	會話劫持攻擊	Session hijacking
	會話重放攻擊	Session replay
	中間人攻擊	Man in the middle
加解密 (Cryptography)	金鑰產生機制或金鑰管理不佳	Poor key generation or key management
	弱的或自定義的加密方法	Weak or custom encryption
	Checksum 偽造	Checksum spoofing
參數竄改 (Parameter Manipulation)	查詢字串操控	Query string manipulation
	表單欄位操控	Form field manipulation
	Cookie 操控	Cookie manipulation
	HTTP 標頭操控	HTTP header manipulation
例外處理 (Exception Management)	攻擊者取得系統細節	Attacker reveals implementation details
	拒絕服務	Denial of service
稽核紀錄 (Auditing and Logging)	使用者否認操作	User denies performing an operation
	攻擊者進行攻擊而不留痕跡	Attackers exploit an application without leaving a trace
	攻擊者掩飾攻擊軌跡	Attackers cover their tracks

資料來源： http://www.microsoft.com - Improving Web Application Security: Threats and Countermeasures

資料流 (EDFD)

- **EDFD** (Extend Data Flow Diagram)
- EDFD 組成元素



資料來源： http://www.microsoft.com - Introduction to SDL Threat Modeling

DFD元素與對應潛在威脅

威脅類型 / 組成元素	S 偽冒	T 竄改	R 否認行為	I 資訊洩漏	D 拒絕存取服務	E 權限提升
外部實體	V		V			
程序	V	V	V	V	V	V
資料流		V		V	V	
資料儲存庫		V	V	V	V	

資料來源： http://www.microsoft.com - Introduction to SDL Threat Modeling

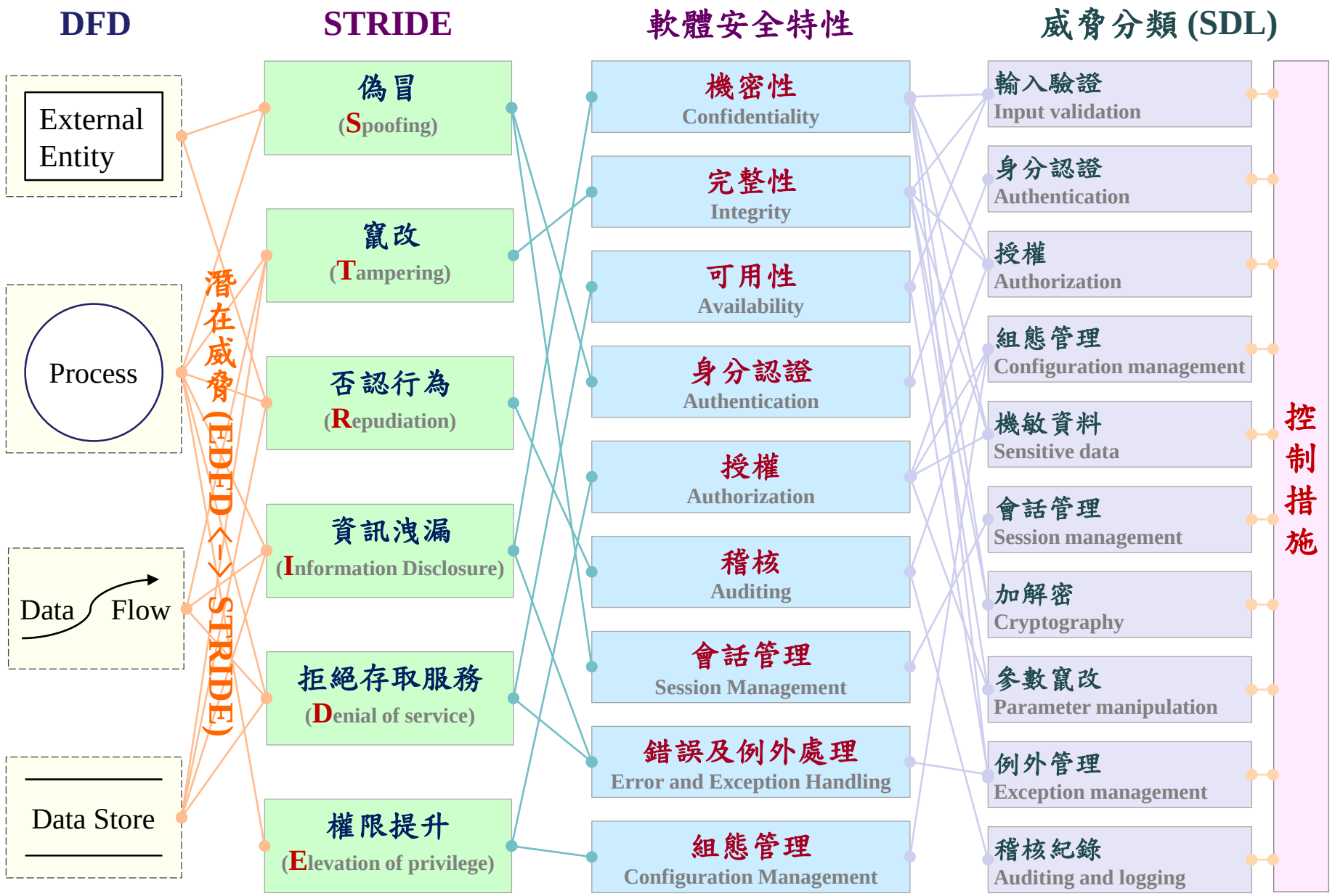
威脅評估 (DREAD)

字首	項目	說明	高(3 分)	中(2 分)	低(1 分)
D	可能傷害 Damage potential	如果此威脅發生，傷害有多大?	攻擊者取得完整的機敏資料;取得系統最高權限	洩漏部分機敏之資料	洩漏系統細節之資訊
R	可重製性 Reproducibility	重新產生針對此一威脅的攻擊的困難程度?	攻擊可以不停重製且不需要暫停時間	攻擊可以重製，但需要暫停時間	攻擊難以重製，或需要對弱點有深入知識才能重製
E	可刺探性 Exploitability	發動攻擊的困難程度?	不具有相關技能者也可發動攻擊	具有相關技能者可以發動攻擊	需要對弱點有深入知識才能發動攻擊
A	影響使用者數 Affected users	受影響的使用者人數?	所有的使用者	部分使用者	極低比例使用者
D	可發現性 Discoverability	發現弱點的困難程度?	公開資訊已揭露攻擊手法及漏洞	已被找到相關的資訊，但漏洞尚未被揭露	極少用到的功能且使用者難以找出漏洞

風險(Risk) = 發生可能性(Probability) X 影響(Damage Potential)

= (可重製性+可刺探性+可發現性) X (可能傷害+影響人數)

EDFD與威脅分類(SDL)



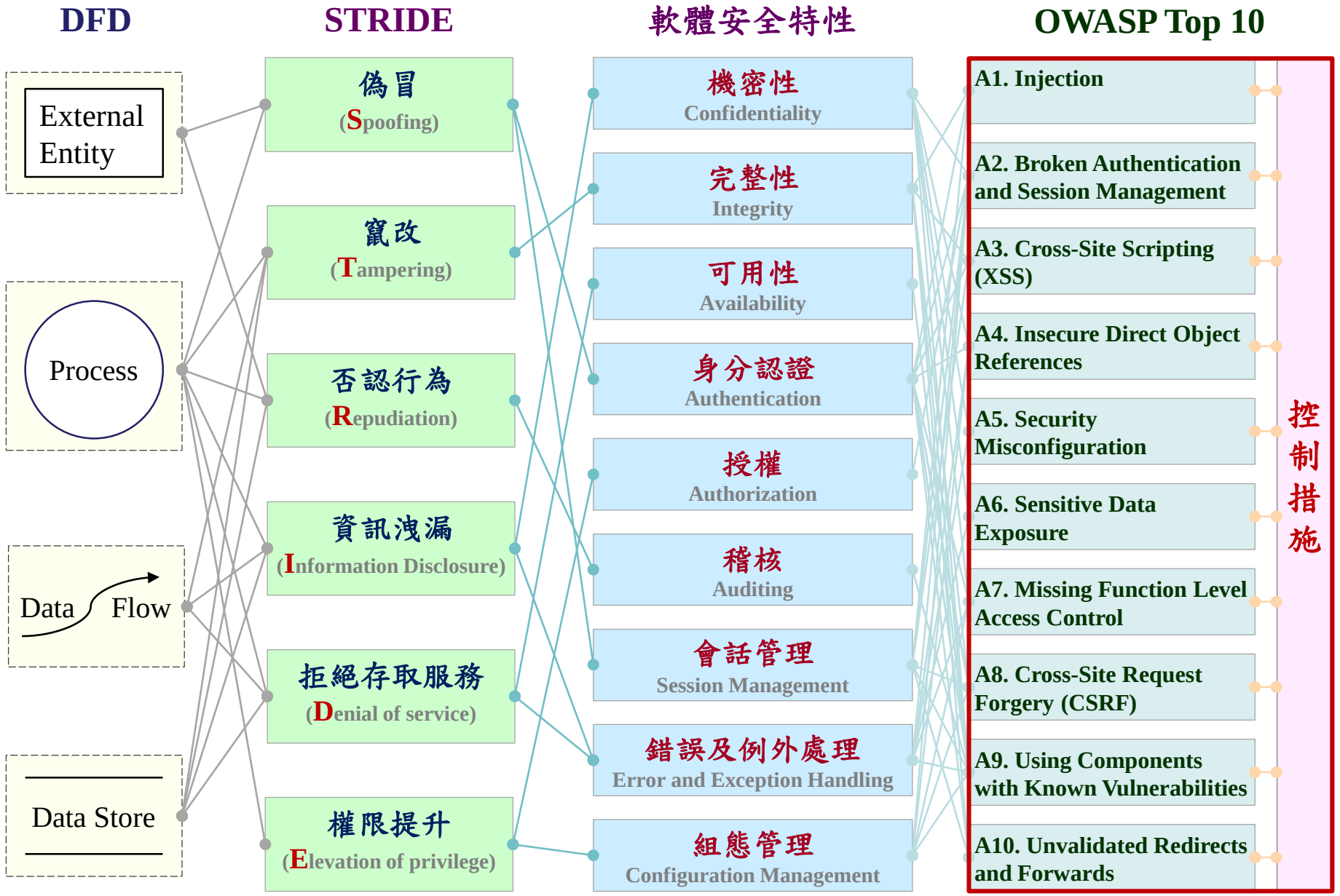
發展降低風險的控制措施

EDFD	STRIDE 潛在威脅	威脅分類	可能威脅	風險值	控制措施
登入驗證功能	偽冒 (Spoofing)	身分驗證 (Authentication)	暴力破解攻擊 (Brute force attacks)	36	使用複雜的高強度密碼，並定期變更
登入驗證功能	偽冒 (Spoofing)	身分驗證 (Authentication)	字典窮舉攻擊 (Dictionary attacks)	36	使用複雜的高強度密碼，並定期變更
登入驗證功能	竄改 (Tampering)	輸入驗證 (Input Validation)	跨站腳本攻擊 (Cross-site scripting)	36	使用白名單方式進行輸入驗證，並使用正規表達式驗證所輸入的資料
登入驗證功能	資訊洩漏 (Information Disclosure)	輸入驗證 (Input Validation)	SQL 注入 (SQL injection)	36	輸入資料需防範SQL Injection攻擊
登入使用者	偽冒 (Spoofing)	身分驗證 (Authentication)	網路竊聽 (Network eavesdropping)	32	使用SSL
登入使用者	偽冒 (Spoofing)	會話管理 (Session Management)	會話劫持攻擊 (Session hijacking)	32	使用內建 session managent
傳回活動列表畫面	竄改 (Tampering)	機密資料 (Sensitive Data)	網路竊聽 (Network Eavesdropping)	30	使用SSL
活動列表功能	竄改 (Tampering)	輸入驗證 (Input Validation)	跨站腳本攻擊 (Cross-site scripting)	30	使用 HTMLEncode 和 URLEncode 函數對輸出進行編碼
登入使用者	否認行為 (Repudiation)	稽核紀錄 (Auditing and Logging)	使用者否認操作 (User denies performing an operation)	27	有Log紀錄
登入驗證功能	否認行為 (Repudiation)	稽核紀錄 (Auditing and Logging)	使用者否認操作 (User denies performing an operation)	27	有Log紀錄
登入驗證功能	資訊洩漏 (Information Disclosure)	例外處理 (Exception Management)	攻擊者取得系統細節 (Attacker reveals implementation details)	27	系統錯誤訊息不得顯示於客戶端
請求活動列表	竄改 (Tampering)	機密資料 (Sensitive Data)	網路竊聽 (Network Eavesdropping)	25	使用SSL
傳回活動列表畫面	拒絕存取服務 (Denial of service)	例外處理 (Exception Management)	攻擊者取得系統細節 (Attacker reveals implementation details)	24	系統錯誤訊息不得顯示於客戶端
活動列表功能	權限提升 (Elevation of privilege)	授權 (Authorization)	權限提升 (Elevation of privilege)	24	使用最小權限
使用者資料表	資訊洩漏 (Information Disclosure)	機密資料 (Sensitive Data)	存取敏感資料 (Access to sensitive data in storage)	24	加密
活動列表功能	偽冒 (Spoofing)	身分驗證 (Authentication)	網路竊聽 (Network eavesdropping)	21	使用SSL
活動列表功能	否認行為 (Repudiation)	稽核紀錄 (Auditing and Logging)	使用者否認操作 (User denies performing an operation)	20	有Log紀錄
使用者資料表	否認行為 (Repudiation)	稽核紀錄 (Auditing and Logging)	攻擊者進行攻擊而不留痕跡 (Attackers exploit an application without leaving a	18	

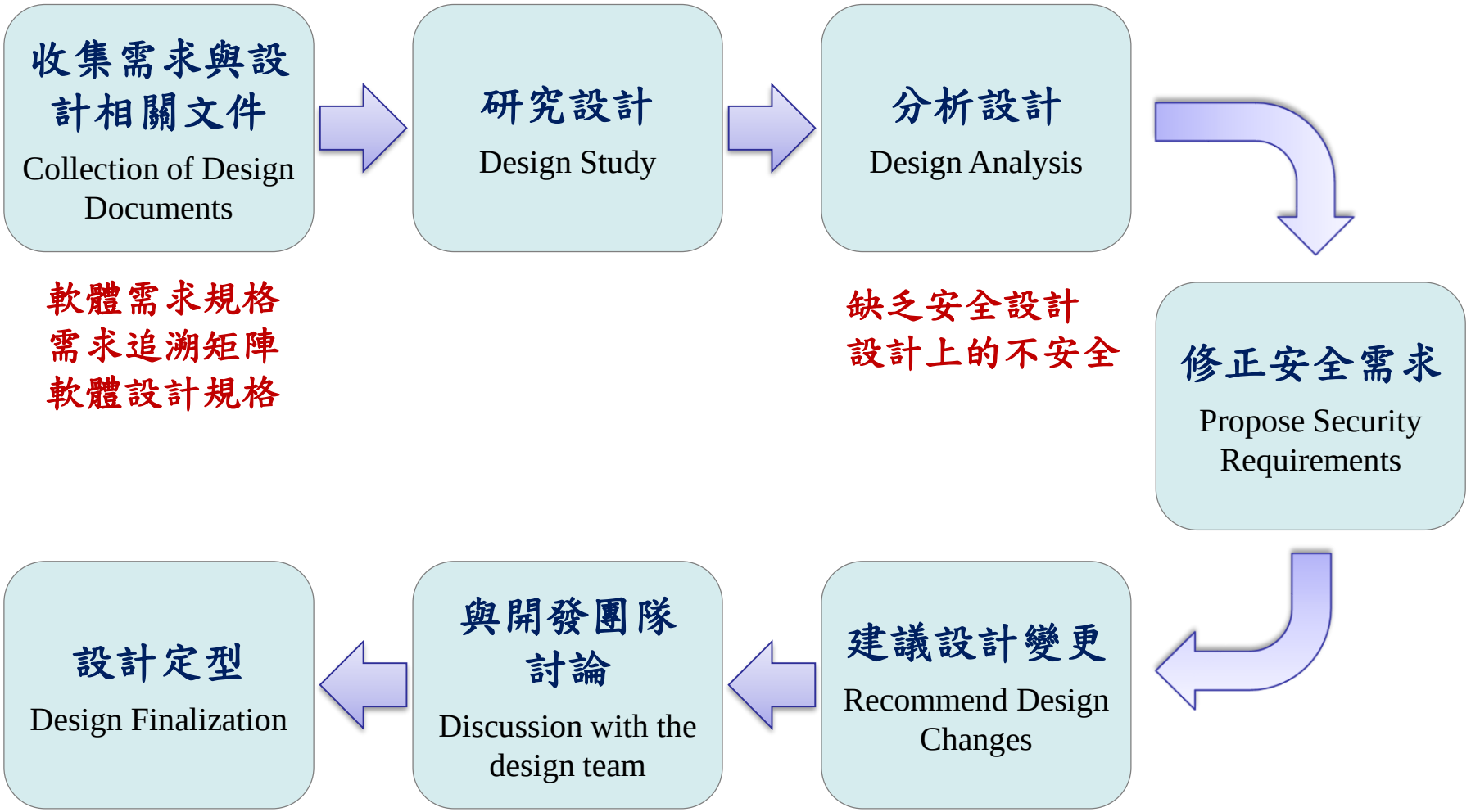
STRIDE 與 OWASP Top 10

STRIDE / OWASP Top 10	(S) 偽冒	(T) 竄改	(R) 否認行為	(I) 資訊洩漏	(D) 拒絕存取服務	(E) 權限提升	軟體安全特性
A1 - Injection	●	●					完整性、身分認證、會話管理
A2 - Broken Auth. & Session Management	●			●		●	機密性、身分認證、授權、會話管理、錯誤及例外處理、組態管理
A3 - Cross-Site Scripting (XSS)	●	●					完整性、身分認證、會話管理
A4 - Insecure Object References				●		●	機密性、授權、錯誤及例外處理、組態管理
A5 - Security Misconfiguration				●			機密性、錯誤及例外處理
A6 - Sensitive Data Exposure				●			機密性、錯誤及例外處理
A7 - Missing Function Level Access Control				●		●	機密性、授權、錯誤及例外處理、組態管理
A8 - Cross Site Request Forgery (CSRF)	●	●				●	完整性、身分認證、授權、會話管理、組態管理
A9 - Using Components with Known Vuln.	●	●	●	●	●	●	機密性、完整性、可用性、身分認證、授權、會話管理、稽核、錯誤及例外處理、組態管理
A10 - Unvalidated Redirects and Forwards	●	●					完整性、身分認證、會話管理

EDFD與威脅分類(OWASP Top 10)



安全設計審查



安全軟體開發

安全軟體開發程序

常見威脅

由設計階段進行之風險分析評估結果找出威脅，發展控制措施與安全特點

常見弱點

彙整OWASP、 CWE/SANS 、弱點資料庫...等常見弱點與防禦方法

軟體安全特性

機密性	完整性	可用性
身分認證	授權	稽核
會話管理	錯誤及例外處理	組態管理

程式撰寫規範
Secure Coding Style

程式開發：避免軟體常見漏洞及實作必要控制措施

源碼安全檢查

CSTI 資安科技研究所 CyberSecurity Technology Institute

安全開發環境

項目	說明
軟體或系統處理、儲存、傳輸資料的敏感性	在開發或測試環境中存取使用機敏資料，需有保護措施及管制程序
適用的內部及外部安全需求	包含政策與法規遵循與軟體安全需求來源所列項目
組織支援軟體開發的安全控制措施	包含管理面、實體與技術的控制措施
軟體開發工作環境中人員的可信賴度	人員進用前，對其進行背景查核
軟體發展的委外程度	需確保相關資安風險有效識別分析，控制措施有效運作
區分軟體發展環境	開發、測試以及正式作業環境應作區分，以避免資料混用以及源碼版本的問題
控制開發環境的存取	限制對程式碼、高權限的工具程式(例如後臺管理工具)以及重要資訊的存取
監控軟體發展環境及其中源碼的異動	開發、測試以及正式作業環境中，程式碼版本的變動應受到控管，並記錄其異動
備份另行儲存在安全地點	重要資料或軟體系統應定期備份，另行儲存在非正式作業環境外的地點
控制不同環境間資料的複製移動	開發與測試環境應避免使用正式環境的資料，並建立程序管制不同環境間 資料移動

資料來源：行政院國家資通安全會報技術服務中心-安全軟體發展流程指引

OWASP

- **OWASP** (Open Web Application Security Project)

  OWASP是一個致力於設想，開發，獲取，操作和維護可信任軟體應用程序的組織。

- 主要的安全專案
  - **OWASP Top 10** (十大網路應用系統安全弱點)
  - **OWASP Mobile Top 10** (十大行動裝置安全弱點)
  - **ASVS (Application Security Verification Standard)** (應用程式安全驗證標準)
  - **SAMM (Software Assurance Maturity Model)** (軟體安全成熟度)

資料來源： http://www.owasp.org

OWASP TOP 10 2013 與 2017 RC1

OWASP Top 10 – 2013	OWASP Top 10 – 2017 RC1
A1 – Injection 注入攻擊	**A1 – Injection** 注入攻擊
A2 – Broken Authentication and Session Management 失效的驗證與連線管理	**A2 – Broken Authentication and Session Management** 失效的驗證與連線管理
A3 – Cross-Site Scripting (XSS) 跨腳本攻擊	**A3 – Cross-Site Scripting (XSS)** 跨腳本攻擊
A4 – Insecure Direct Object References 不安全的物件參考 [Merged with A7 A4 – Broken Access Control (Original category in 2003/2004)]	**A4 – Broken Access Control** 失效的存取控制存取控制 [Original category in 2003/2004]
A5 – Security Misconfiguration 不當的安全組態設定	**A5 – Security Misconfiguration** 不當的安全組態設定
A6 – Sensitive Data Exposure 敏感資料曝露	**A6 – Sensitive Data Exposure** 敏感資料曝露
A7 – Missing Function Level Access Control 存取控制缺乏權限分級功能 [Merged with A4]	**A7 – Insufficient Attack Protection** 攻擊保護不足
A8 – Cross-Site Request Forgery (CSRF) 跨站冒名請求	**A8 – Cross-Site Request Forgery (CSRF)** 跨站冒名請求
A9 – Using Components with Known Vulnerabilities 使用已知漏洞元件	**A9 – Using Components with Known Vulnerabilities** 使用已知漏洞元件
A10 – Unvalidated Redirects and Forwards 未經驗證的重新導向與轉送 [Dropped]	**A10 – Underprotected APIs** 未受保護的 API

OWASP Top 10 – 2013 弱點描述範例

A1 Injection

Threat Agents	Attack Vectors	Security Weakness		Technical Impacts	Business Impacts
Application Specific	**Exploitability EASY**	**Prevalence COMMON**	**Detectability AVERAGE**	**Impact SEVERE**	**Application / Business Specific**
Consider anyone who can send untrusted data to the system, including external users, internal users, and administrators.	Attacker sends simple text-based attacks that exploit the syntax of the targeted interpreter. Almost any source of data can be an injection vector, including internal sources.	Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or NoSQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code, but frequently hard to discover via testing. Scanners and fuzzers can help attackers find injection flaws.		Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover.	Consider the business value of the affected data and the platform running the interpreter. All data could be stolen, modified, or deleted. Could your reputation be harmed?

Am I Vulnerable To Injection?

The best way to find out if an application is vulnerable to injection is to verify that all use of interpreters clearly separates untrusted data from the command or query. For SQL calls, this means using bind variables in all prepared statements and stored procedures, and avoiding dynamic queries.

Checking the code is a fast and accurate way to see if the application uses interpreters safely. Code analysis tools can help a security analyst find the use of interpreters and trace the data flow through the application. Penetration testers can validate these issues by crafting exploits that confirm the vulnerability.

Automated dynamic scanning which exercises the application may provide insight into whether some exploitable injection flaws exist. Scanners cannot always reach interpreters and have difficulty detecting whether an attack was successful. Poor error handling makes injection flaws easier to discover.

How Do I Prevent Injection?

Preventing injection requires keeping untrusted data separate from commands and queries.

1. The preferred option is to use a safe API which avoids the use of the interpreter entirely or provides a parameterized interface. Be careful with APIs, such as stored procedures, that are parameterized, but can still introduce injection under the hood.
2. If a parameterized API is not available, you should carefully escape special characters using the specific escape syntax for that interpreter. OWASP's ESAPI provides many of these escaping routines.
3. Positive or "white list" input validation is also recommended, but is not a complete defense as many applications require special characters in their input. If special characters are required, only approaches 1. and 2. above will make their use safe. OWASP's ESAPI has an extensible library of white list input validation routines.

Example Attack Scenarios

Scenario #1: The application uses untrusted data in the construction of the following vulnerable SQL call:

```
String query = "SELECT * FROM accounts WHERE
custID='" + request.getParameter("id") + "'";
```

Scenario #2: Similarly, an application's blind trust in frameworks may result in queries that are still vulnerable, (e.g., Hibernate Query Language (HQL)):

```
Query HQLQuery = session.createQuery("FROM accounts
WHERE custID='" + request.getParameter("id") + "'");
```

In both cases, the attacker modifies the 'id' parameter value in her browser to send: **' or '1'='1**. For example:

http://example.com/app/accountView?id=' or '1'='1

This changes the meaning of both queries to return all the records from the accounts table. More dangerous attacks could modify data or even invoke stored procedures.

References

OWASP

- OWASP SQL Injection Prevention Cheat Sheet
- OWASP Query Parameterization Cheat Sheet
- OWASP Command Injection Article
- OWASP XML eXternal Entity (XXE) Reference Article
- ASVS: Output Encoding/Escaping Requirements (V6)
- OWASP Testing Guide: Chapter on SQL Injection Testing

External

- CWE Entry 77 on Command Injection
- CWE Entry 89 on SQL Injection
- CWE Entry 564 on Hibernate Injection

風險評估 (使用OWASP Risk Rating Methodology)

判斷是否存在弱點

防範方法

攻擊情境範例

參考資料

資料來源： http://www.owasp.org – OWASP Top 10 2013

弱點資料庫 (Vulnerability Database) (1/5)

資料來源： https://nvd.nist.gov

源碼安全檢查

- 源碼安全檢查是對軟體的源碼進行稽核的流程，以驗證正確的安全控制措施已被實作，且如預期般地在正確的地方被使用。
- 源碼安全檢查是確保軟體開發者遵循程式撰寫規範(Coding Style)，採用安全的實作方法，以避免常見的安全弱點。
- 以人工進行源碼安全檢查的方式，可以找出軟體中除錯用源碼、維護用後門與測試帳號等安全危害。
- **源碼安全查檢表**

源碼安全查檢表範例

檢核項目	檢核結果		
1. 資料驗證(DataValidation)	是	否	不適用
1.1 確認驗證資料合法性的機制存在。			
1.2 確認所有可能被惡意使用者修改的輸入處，其輸入資料都被正確的驗證。			
1.3 確認所有輸入資料的正確長度檢查存在。			
1.4 確認輸入欄位、客戶端暫存、HTTP 協定標頭皆有被驗證。			
1.5 確認資料的格式正確且包含的是合法的字元。			
1.6 確認資料驗證機制存在於伺服器端。			
1.7 檢查資料驗證是否確實發生。			
1.8 檢查資料驗證機制其相關程式碼，沒有後門或惡意代碼存在。			
2. 身分認證(Authentication)	是	否	不適用
2.1 確認所有內部與外部的連線(使用者及元件)有適當而足夠的驗證機制。			
2.2 確認所有非公開的功能頁面強制要求認證。			
2.3 若採用 HTTP 協定，確認身分認證的憑據或其他機敏資料是透過 HTTPPOST 方法傳送。			
2.4 任何不在認證機制管控範圍內的功能頁面，已完成檢視評估沒有安全危害可能。			
2.5 確認身分認證的憑據不會以明文方式傳送，採用加密方式傳送			
2.6 確認跳過身分認證的測試用、除錯用後門，沒有存在正式上線的程式碼中			

資料來源：行政院國家資通安全會報技術服務中心-安全軟體發展流程指引

安全軟體測試

安全軟體測試程序

安全軟體測試

安全性測試

以攻擊者的角度去確認軟體承受攻擊的能力，著重在找出軟體的安全錯誤(Bug)與缺陷(Flaw)。

靜態分析

在不執行軟體的狀況下，針對軟體的內容進行分析，比對已知弱點模式，找出可能的缺陷或錯誤。

源碼檢測
Checkmarx、
Fortify SCA…

動態分析

將軟體在實際的環境中執行起來並進行分析的活動。

弱點掃描
Acunetix 、 Nessus 、
WebInspect…

安全需求驗證

測試軟體中安全相關功能是否符合需求並運作正確

OWASP ASVS

安全軟體部署

安全軟體部署與組態設定原則

作業系統

- 定期 Patch 更新，重要伺服器應於測試環境測試完畢後再行更新。
- 關閉不使用的服務及 Port。
- 正確的安全設定 。
- 注意安全警示(Security Alert)網站之通告。

網頁或應用程式伺服器

- 定期評估是否更新伺服器版本。
- 安全限制。
- 認證方法。
- 錯誤處理頁面。
- Session Timeout時間。

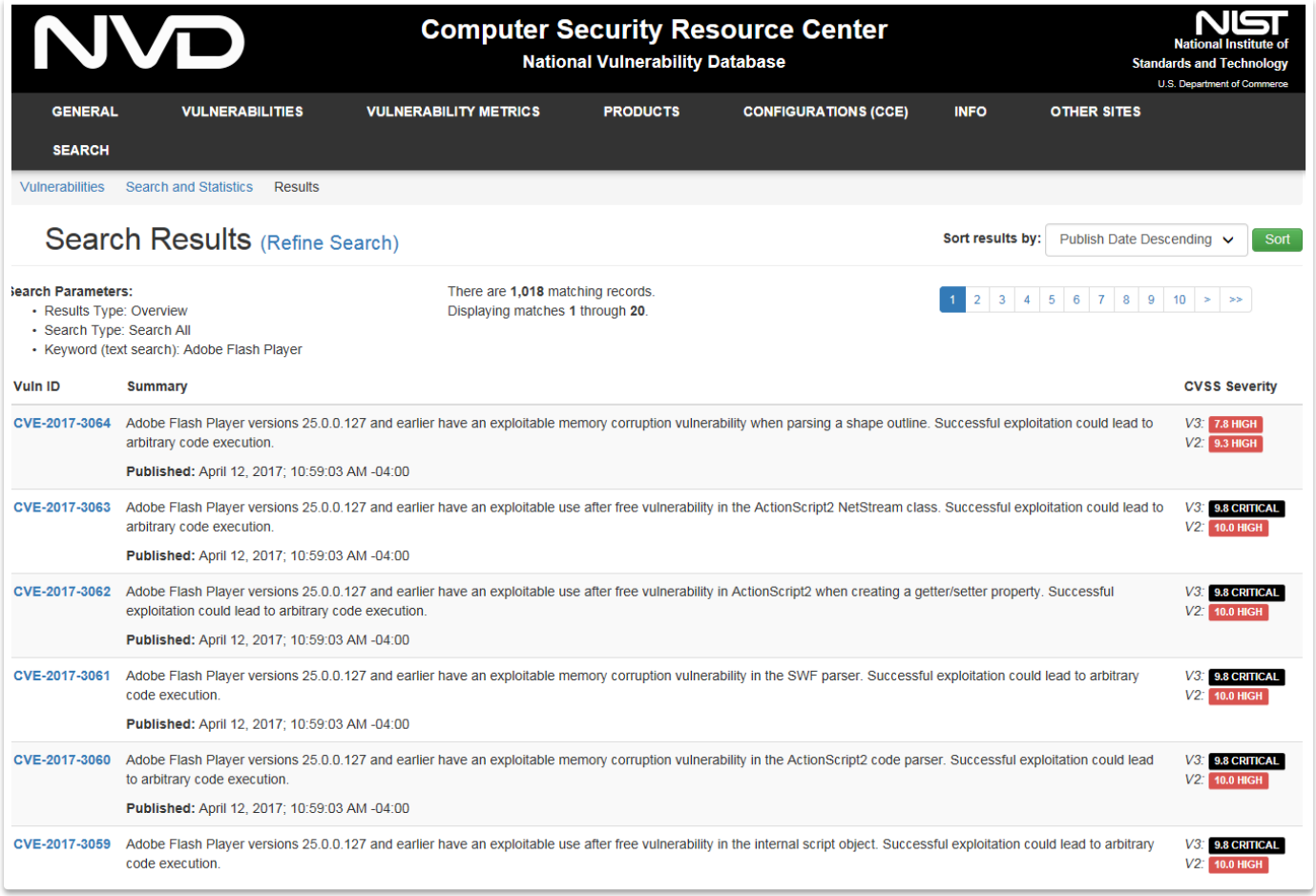
資料庫

- 不使用預設帳號密碼。
- 不使用過高的管理權限連接資料庫。
- 前後台(一般功能與管理功能)不使用相同帳號連接資料庫。
- 定期更新資料庫的安全修補。
- 盡可能採用加密連線。
- 存取行為進行監控，定期稽核

軟體本身

- 敏感資料(例如帳號密碼)不以明文方式放置於設定檔或源碼中。
- 移除除錯用的源碼。
- 移除維護用的後門。
- 移除測試帳號。
- 不開啟底層非必要的服務。
- 以所需的最低權限執行軟體。

弱點公告網站 1/3



NVD

Computer Security Resource Center
National Vulnerability Database

NIST National Institute of Standards and Technology U.S. Department of Commerce

GENERAL | VULNERABILITIES | VULNERABILITY METRICS | PRODUCTS | CONFIGURATIONS (CCE) | INFO | OTHER SITES | SEARCH

Vulnerabilities | Search and Statistics | Results

Search Results (Refine Search)

Sort results by: Publish Date Descending Sort

Search Parameters:
- Results Type: Overview
- Search Type: Search All
- Keyword (text search): Adobe Flash Player

There are **1,018** matching records.
Displaying matches **1** through **20**.

1 2 3 4 5 6 7 8 9 10 > >>

Vuln ID	Summary	CVSS Severity
CVE-2017-3064	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable memory corruption vulnerability when parsing a shape outline. Successful exploitation could lead to arbitrary code execution. **Published:** April 12, 2017; 10:59:03 AM -04:00	V3: 7.8 HIGH V2: 9.3 HIGH
CVE-2017-3063	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable use after free vulnerability in the ActionScript2 NetStream class. Successful exploitation could lead to arbitrary code execution. **Published:** April 12, 2017; 10:59:03 AM -04:00	V3: 9.8 CRITICAL V2: 10.0 HIGH
CVE-2017-3062	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable use after free vulnerability in ActionScript2 when creating a getter/setter property. Successful exploitation could lead to arbitrary code execution. **Published:** April 12, 2017; 10:59:03 AM -04:00	V3: 9.8 CRITICAL V2: 10.0 HIGH
CVE-2017-3061	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable memory corruption vulnerability in the SWF parser. Successful exploitation could lead to arbitrary code execution. **Published:** April 12, 2017; 10:59:03 AM -04:00	V3: 9.8 CRITICAL V2: 10.0 HIGH
CVE-2017-3060	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable memory corruption vulnerability in the ActionScript2 code parser. Successful exploitation could lead to arbitrary code execution. **Published:** April 12, 2017; 10:59:03 AM -04:00	V3: 9.8 CRITICAL V2: 10.0 HIGH
CVE-2017-3059	Adobe Flash Player versions 25.0.0.127 and earlier have an exploitable use after free vulnerability in the internal script object. Successful exploitation could lead to arbitrary code execution.	V3: 9.8 CRITICAL V2: 10.0 HIGH

美國國家弱點資料庫(National Vulnerability Database)
https://nvd.nist.gov/

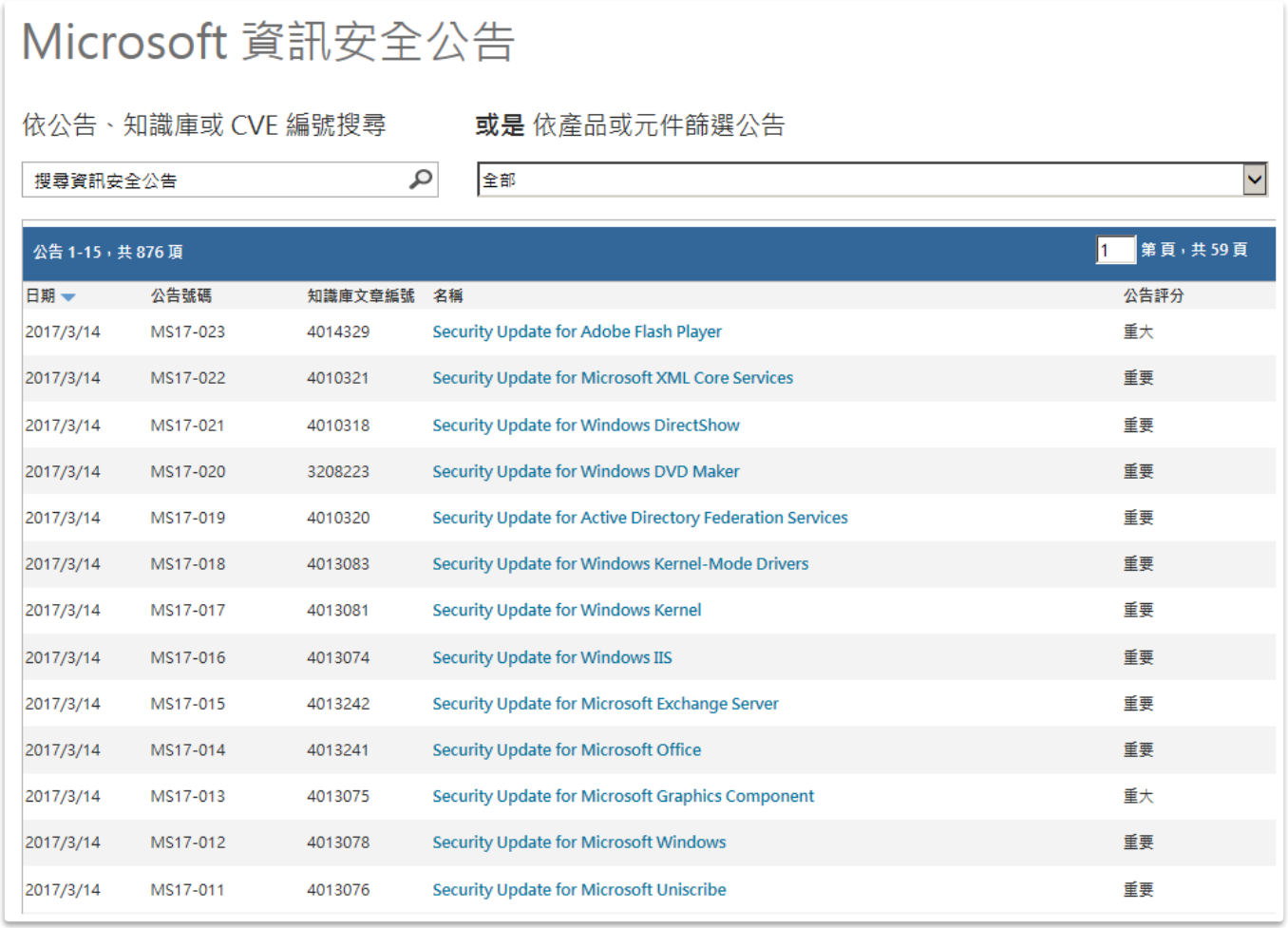
CSTI 資安科技研究所 CyberSecurity Technology Institute

弱點公告網站 2/4

行政院國家資通安全會報技術服務中心(NCCST) 漏洞公告
https://www.nccst.nat.gov.tw/Vulnerability

CSTI 資安科技研究所
CyberSecurity Technology Institute

弱點公告網站 3/4



Microsoft 資訊安全公告

依公告、知識庫或 CVE 編號搜尋　　**或是** 依產品或元件篩選公告

搜尋資訊安全公告　　全部

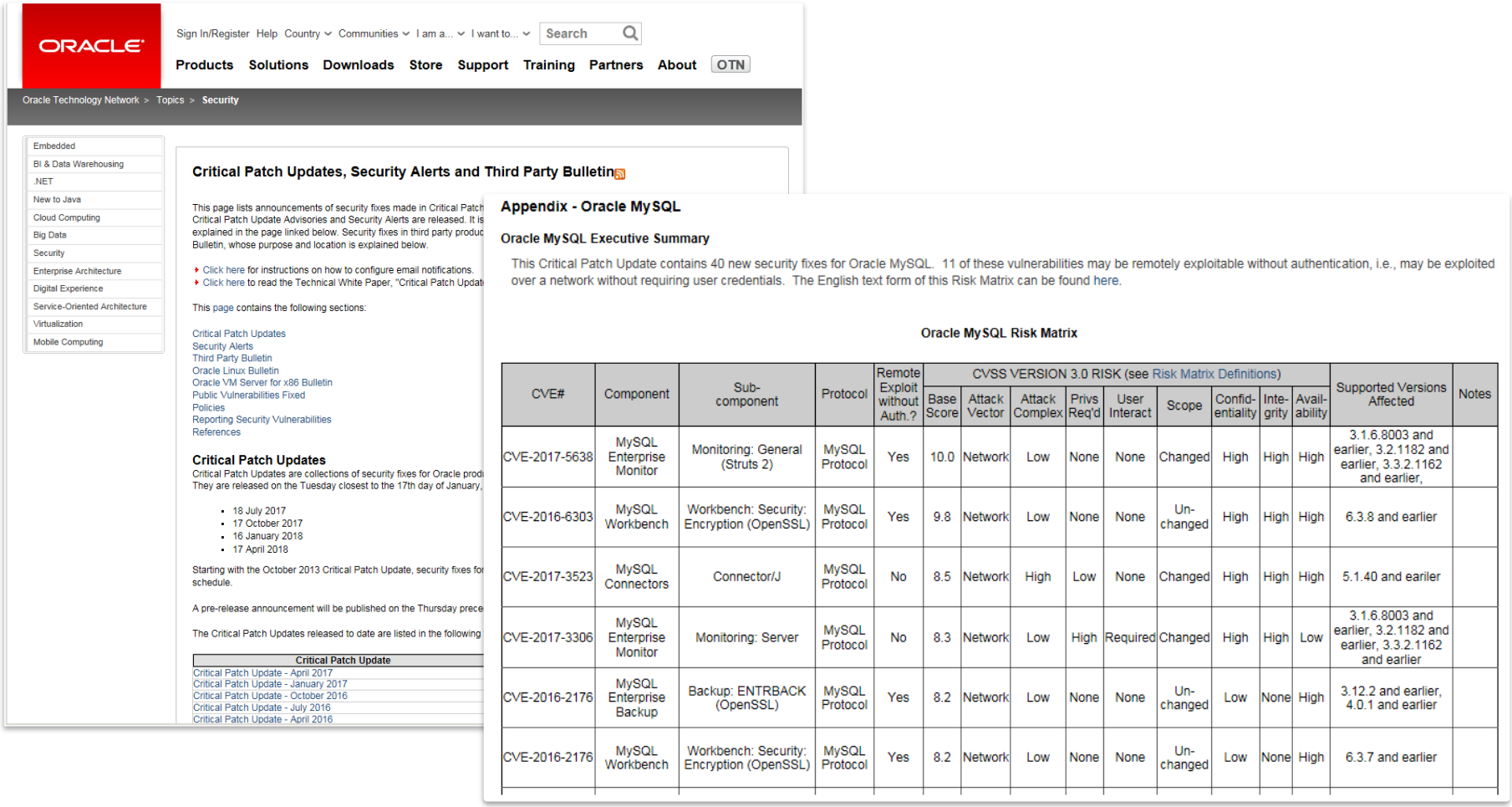
公告 1-15，共 876 項　　1 第頁，共 59 頁

日期	公告號碼	知識庫文章編號	名稱	公告評分
2017/3/14	MS17-023	4014329	Security Update for Adobe Flash Player	重大
2017/3/14	MS17-022	4010321	Security Update for Microsoft XML Core Services	重要
2017/3/14	MS17-021	4010318	Security Update for Windows DirectShow	重要
2017/3/14	MS17-020	3208223	Security Update for Windows DVD Maker	重要
2017/3/14	MS17-019	4010320	Security Update for Active Directory Federation Services	重要
2017/3/14	MS17-018	4013083	Security Update for Windows Kernel-Mode Drivers	重要
2017/3/14	MS17-017	4013081	Security Update for Windows Kernel	重要
2017/3/14	MS17-016	4013074	Security Update for Windows IIS	重要
2017/3/14	MS17-015	4013242	Security Update for Microsoft Exchange Server	重要
2017/3/14	MS17-014	4013241	Security Update for Microsoft Office	重要
2017/3/14	MS17-013	4013075	Security Update for Microsoft Graphics Component	重大
2017/3/14	MS17-012	4013078	Security Update for Microsoft Windows	重要
2017/3/14	MS17-011	4013076	Security Update for Microsoft Uniscribe	重要

Microsoft 資訊安全公告

https://technet.microsoft.com/zh-tw/security/bulletins/

弱點公告網站 4/4



CVE#	Component	Sub-component	Protocol	Remote Exploit without Auth.?	CVSS VERSION 3.0 RISK (see Risk Matrix Definitions)									Supported Versions Affected	Notes
					Base Score	Attack Vector	Attack Complex	Privs Req'd	User Interact	Scope	Confid-entiality	Inte-grity	Avail-ability		
CVE-2017-5638	MySQL Enterprise Monitor	Monitoring: General (Struts 2)	MySQL Protocol	Yes	10.0	Network	Low	None	None	Changed	High	High	High	3.1.6.8003 and earlier, 3.2.1182 and earlier, 3.3.2.1162 and earlier,	
CVE-2016-6303	MySQL Workbench	Workbench: Security: Encryption (OpenSSL)	MySQL Protocol	Yes	9.8	Network	Low	None	None	Un-changed	High	High	High	6.3.8 and earlier	
CVE-2017-3523	MySQL Connectors	Connector/J	MySQL Protocol	No	8.5	Network	High	Low	None	Changed	High	High	High	5.1.40 and eariler	
CVE-2017-3306	MySQL Enterprise Monitor	Monitoring: Server	MySQL Protocol	No	8.3	Network	Low	High	Required	Changed	High	High	Low	3.1.6.8003 and earlier, 3.2.1182 and earlier, 3.3.2.1162 and earlier	
CVE-2016-2176	MySQL Enterprise Backup	Backup: ENTRBACK (OpenSSL)	MySQL Protocol	Yes	8.2	Network	Low	None	None	Un-changed	Low	None	High	3.12.2 and earlier, 4.0.1 and earlier	
CVE-2016-2176	MySQL Workbench	Workbench: Security: Encryption (OpenSSL)	MySQL Protocol	Yes	8.2	Network	Low	None	None	Un-changed	Low	None	High	6.3.7 and earlier	

Oracle 重大更新公告

https://www.oracle.com/technetwork/topics/security/whatsnew/index.html

CSTI 資安科技研究所 CyberSecurity Technology Institute

變更管理

識別可能的變更	- 因遭遇軟體錯誤或要求新功能而產生變更的需求。 - 提出變更要求。
分析變更請求	- 分析變更的技術可行性。 - 分析成本與效益。
評估變更	變更管理者根據可行性、成本與效益決定是否進行變更。
計畫變更	- 分析變更的影響，包含對「軟體安全性」的影響。 - 對變更進行計畫。
實作變更	- 根據計畫實作變更。 - 更新相關文件，含「軟體安全性」。 - 測試變更。 - 釋出變更。反應變更需求的版本上線。
檢視與結案	- 驗證變更生效。 - 執行結案程序。

Secure Socket Layer (SSL)

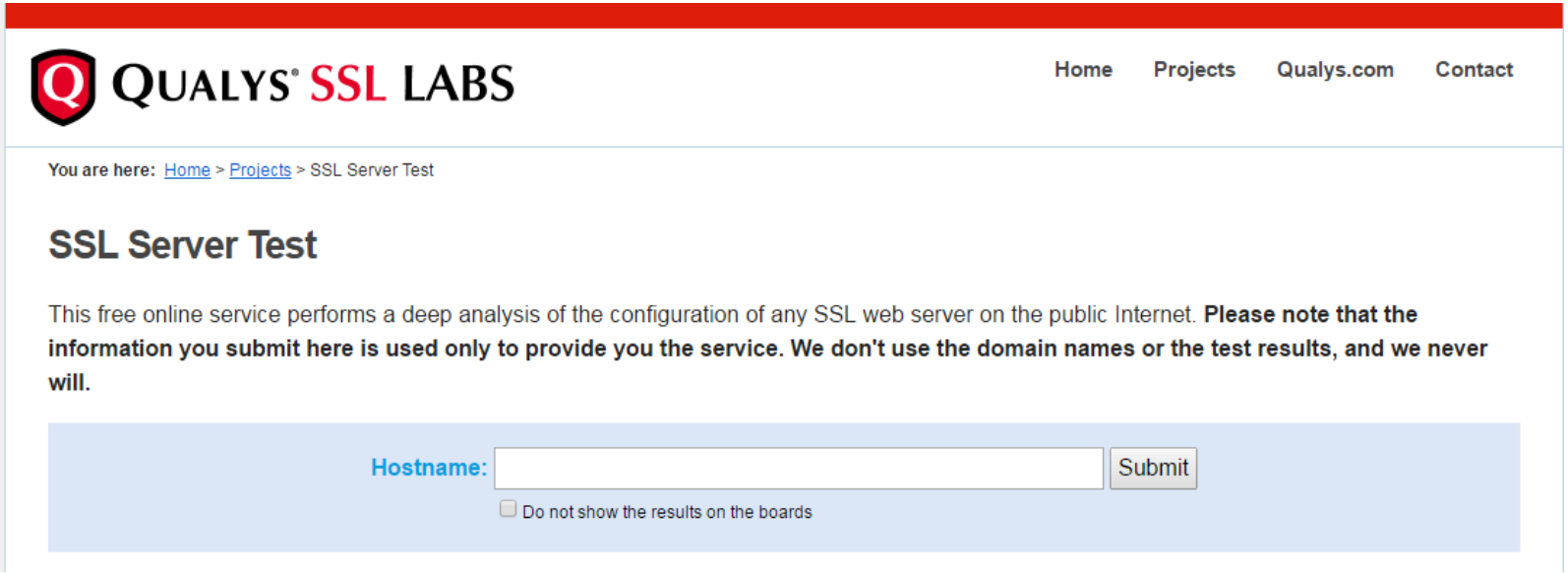
- 使用安全的協定
  - Secure Sockets Layer (SSL)：SSLv2、SSLv3 不安全
  - Transport Layer Security (TLS)：
    - TLSv1.0 不安全
    - TLSv1.1 和 TLSv1.2 仍未發現重大安全性問題
- 從可信賴的 CA 取得證書
  - https://wiki.mozilla.org/CA:IncludedCAs (Mozilla)
  - https://support.apple.com/zh-tw/HT207177 (Apple iOS 10)
  - …

SSLv3 加密協定存在中間人攻擊弱點，弱點編號 CVE-2014-3566 (POODLE)

近期美國國家標準技術研究所(NIST)的國家弱點資料庫(NVD)發布弱點編號CVE-2014-3566 (POODLE) [1-5]，透過該弱點駭客可在客戶端與伺服器兩端均使用SSLv3加密協定建立連線時進行中間人攻擊，藉由攔截與修改HTTPS封包資訊的方式，嘗試向伺服器主機建立連線，進而獲取使用者的相關傳輸數據與機敏資訊(cookies, and/or authorization header contents)。

為確保平台安全性，請各機關確認所屬瀏覽器設定與系統平台(工作站主機及伺服器)所使用的SSL/TLS版本資訊，建議停用SSLv3支援選項改以TLSv1.2作為加密協定。如系統平台不支援TLS加密連線機制，請儘速更新修補SSL相關套件。

網站 SSL 安全性檢測



Qualys SSL LABS

https://www.ssllabs.com/ssltest/

參考資料來源

- 資訊系統安全等級

  行政院國家資通安全會報-資訊系統分級與資安防護基準作業規定 (104年7月修正)

  http://www.nicst.ey.gov.tw/News_Content.aspx?n=626B7A2643794AB0&sms=C43ECA251722A365&s=7167B7BF3097234D

- 安全需求查檢表
- 源碼安全查檢表

  行政院國家資通安全會報技術服務中心-安全軟體發展流程指引

  https://www.nccst.nat.gov.tw/CommonSpecification?lang=zh

- OWASP ASVS 3.0.1

  https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

- SDL 威脅與緩解措施

  https://www.microsoft.com/en-us/download/details.aspx?id=1330

參考資料來源

- OWASP Top 10

  https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- CWE/SANS Top 25

  http://cwe.mitre.org/top25/
- WASC Threat Classification

  http://projects.webappsec.org/w/page/13246978/Threat%20Classification

CSTI 資安科技研究所
CyberSecurity Technology Institute