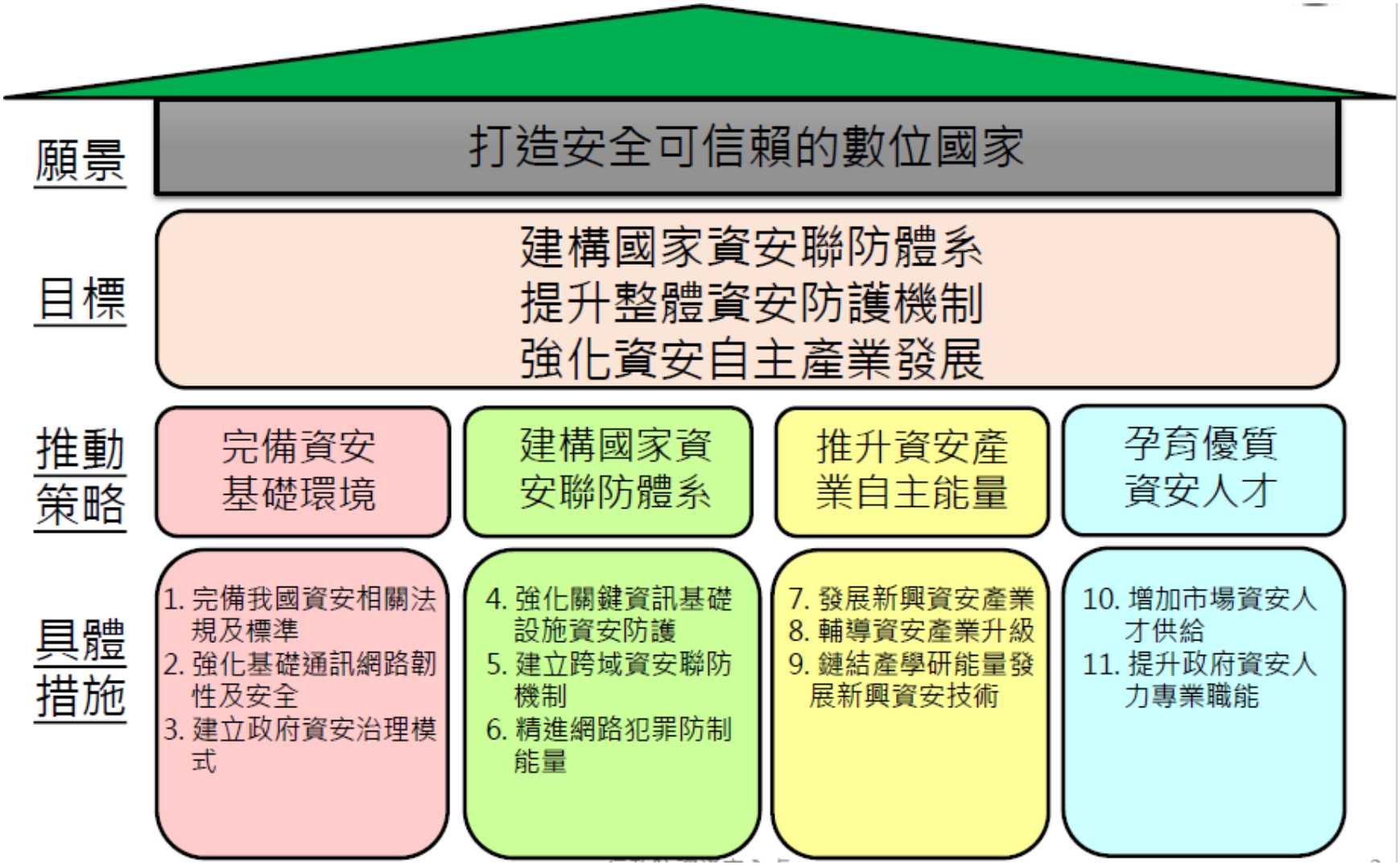
資通安全管理法規範重點宣導

Benny

資訊安全觀念

- 沒有任何一套系統及措施可提供百分之百的安全防護。
- 人是系統安全中最薄弱的一環。
- 便利性及安全性常常互相衝突。

國家資通安全發展方案


願景
打造安全可信賴的數位國家
目標
建構國家資安聯防體系
提升整體資安防護機制
強化資安自主產業發展
推動策略
完備資安基礎環境
建構國家資安聯防體系
推升資安產業自主能量
孕育優質資安人才
具體措施
1. 完備我國資安相關法規及標準
2. 強化基礎通訊網路韌性及安全
3. 建立政府資安治理模式
4. 強化關鍵資訊基礎設施資安防護
5. 建立跨域資安聯防機制
6. 精進網路犯罪防制能量
7. 發展新興資安產業
8. 輔導資安產業升級
9. 鏈結產學研能量發展新興資安技術
10. 增加市場資安人才供給
11. 提升政府資安人力專業職能

關鍵基礎設施(CI)

資安法結構

- 行政院、委託或委任單位、各公務機關
- 中央目的事業主管機關權責
- 權限委託

- **資安責任等級分級**
- 資安維護計畫之制定與實施
- 年度資安維護計畫實施情形提出
- **資安稽核**
- **資安事件通報應變**
- 改善報告
- 公告
- 定期公布國家資通安全情勢報告及資通安全發展方案
- **建立情資分享機制**

- **資安責任等級分級**
- 資安維護計畫之制定與實施
- 年度資安維護計畫實施情形提出
- 資安長設置
- 年度資安維護計畫實施情形提出
- 資安稽核
- 改善報告
- **資安事件通報應變**
- **公務機關人員獎懲標準**

- **公務機關人員獎懲標準**
- 通報義務
- 資安維護計畫實施
- 改善報告
- 應變機制

- **資安責任等級分級**
- 資安維護計畫之制定與實施
- 年度資安維護計畫實施情形提出
- **資安稽核**
- **資安事件通報應變**
- 改善報告
- 公告
- 罰則

公務機關之資通安全管理

✓ 應訂定資通安全維護計畫§9
✓ 應訂定通報及應變機制§13I

行政院

- 應提出年度資通安全維護計畫之實施情形§11
- 應提出改善報告§12 II
- 應通報資通安全事件§13II
- 應提出資通安全事件之調查、處理及改善報告§13III

上級或監督機關 ⇄ 下級或受監督機關

- 應稽核資通安全維護計畫實施情形§12I

- 擘劃並推動國家資通安全政策
- 資通安全科技發展
- 國際交流合作及資通安全整體防護
- 定期公布國家資安情勢報告及資通安全發展方案

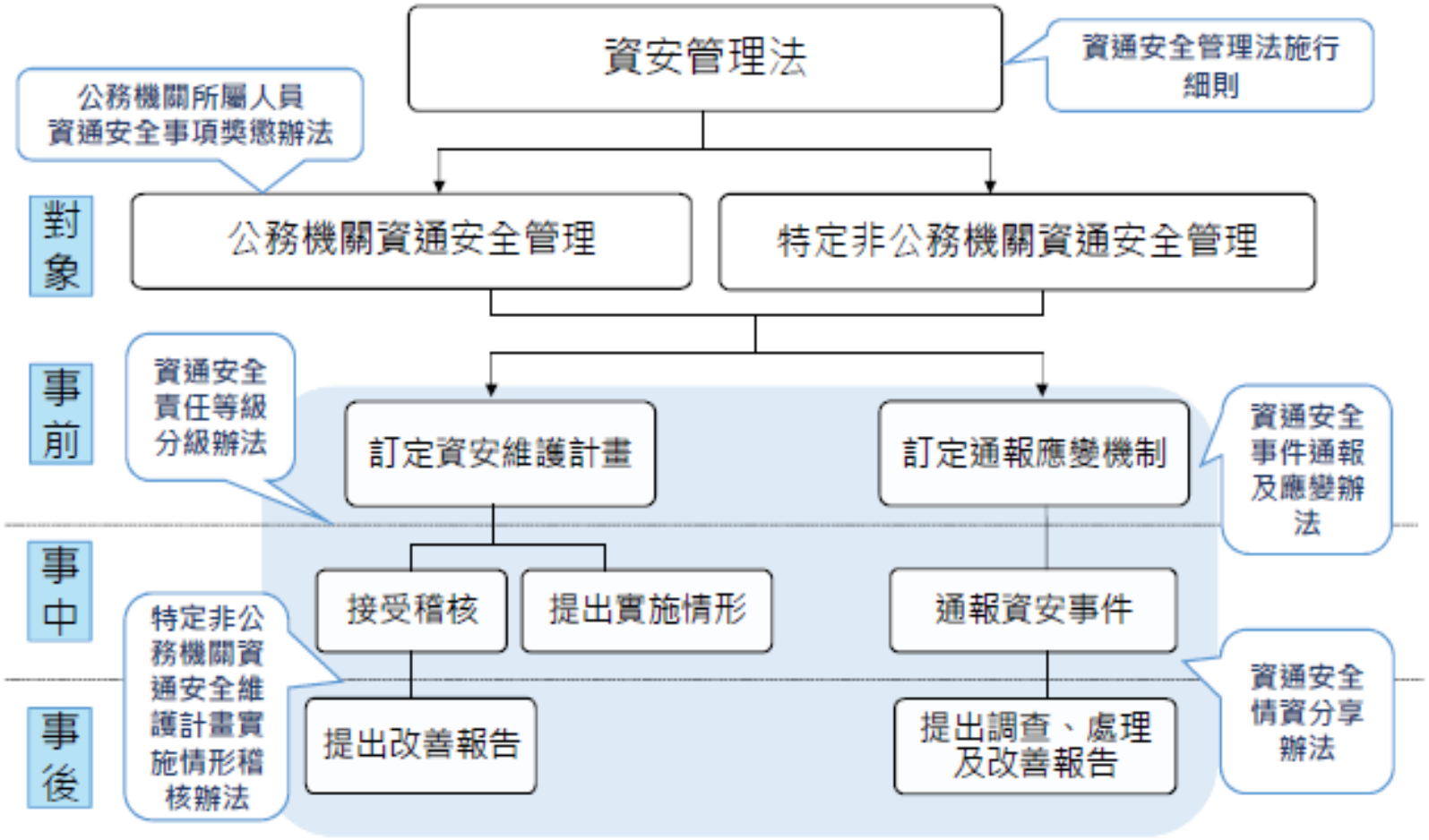
訂定

- ✓資安管理法施行細則§22
- ✓資安責任等級分級辦法§6
- ✓資安事件通報及應變辦法§ 13、17
- ✓維護計畫實施情形稽核辦法§ 6、12
- ✓資安情資分享辦法§7

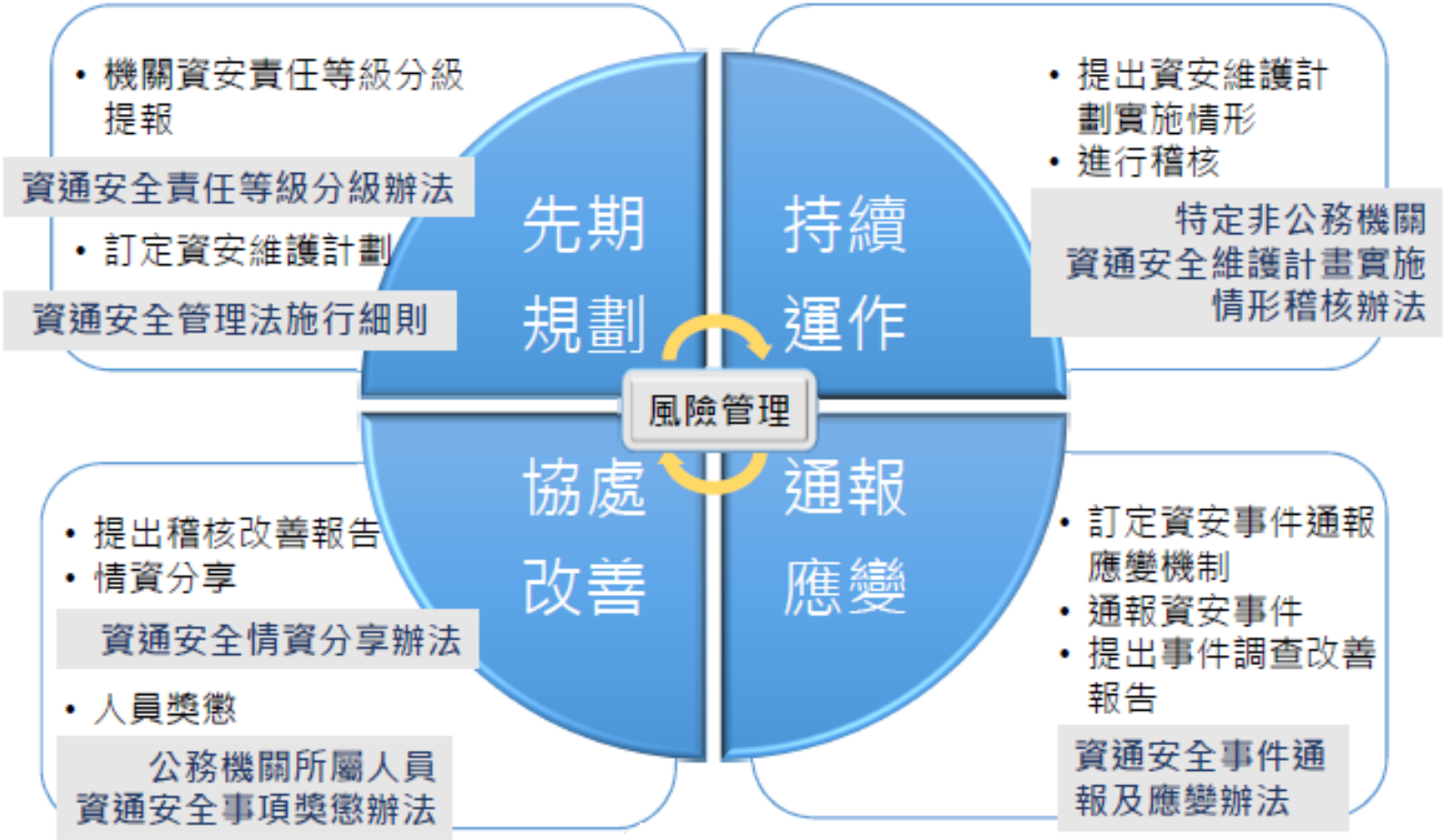
總統府、立法院、司法院、考試院、監察院、直轄市政府、直轄市議會、縣（市）政府及縣（市）議會

設置資通安全長§10

資安法管理架構


資安管理法
資通安全管理法施行細則
公務機關所屬人員資通安全事項獎懲辦法
對象
公務機關資通安全管理
特定非公務機關資通安全管理
事前
資通安全責任等級分級辦法
訂定資安維護計畫
訂定通報應變機制
資通安全事件通報及應變辦法
事中
接受稽核
提出實施情形
通報資安事件
特定非公務機關資通安全維護計畫實施情形稽核辦法
事後
提出改善報告
提出調查、處理及改善報告
資通安全情資分享辦法

以風險管理為核心的資安防護



資安法架構

第1章 總則(§1~§8)	立法目的、名詞定義、資通安全產業之推動、行政院職責、幕僚任務委任或委託、資安責任等級分級、情資分享機制、資通委外監督
第2章 公務機關資通安全管理(§9~§14)	資通安全管理與維護計畫、資通安全長之設置、年度資通安全報告之提出、資通安全查核、通報應變措施、獎懲措施
第3章 非公務機關資通安全管理(§15~§18)	關鍵基礎設施提供者資通安全維護之管理與監督、受指定之非公務機關所提供之產品或服務資通安全管理之管理與監督、資通安全事件通報應變、行政檢查
第4章 罰則(§19~§22)	行政處分
第5章 附則(§23~§24)	施行細則授權、施行日期

資通安全管理法施行細則

資通安全維護計畫內容

基於風險管理之基礎，包含下列內容(13款)

維護計畫實施情形，應包括各款之執行成果與相關說明。
詳本說明會議題三–資通安全維護計畫撰寫作業。

資通系統建置、服務委外辦理注意事項

- 考量委外項目之性質、資通安全需求，選任適當之受託者，並監督其資通安全維護。
  - 資安法施行後, 不論是新開發或是增修，只要有委外就要適用。
  - 考量個案不同，受託者可自行使用第三方軟體進行安全性檢測。
  - 惟如該資通系統屬委託機關之核心資通系統，或委託案件金額在1,000萬元以上，委託機關應自行或另行委託第三方進行安全性檢測。

委外之前

- 受託者應具備完善之資通安全管理措施或通過第三方驗證
- 受託者應配置之資安專業人員(數量、資格、證照、經驗)
- 受託者得否複委託，及進行複委託應注之事項
- 受託業務涉及國家機密者，相關執行人員應接受適任性查核

委外之後

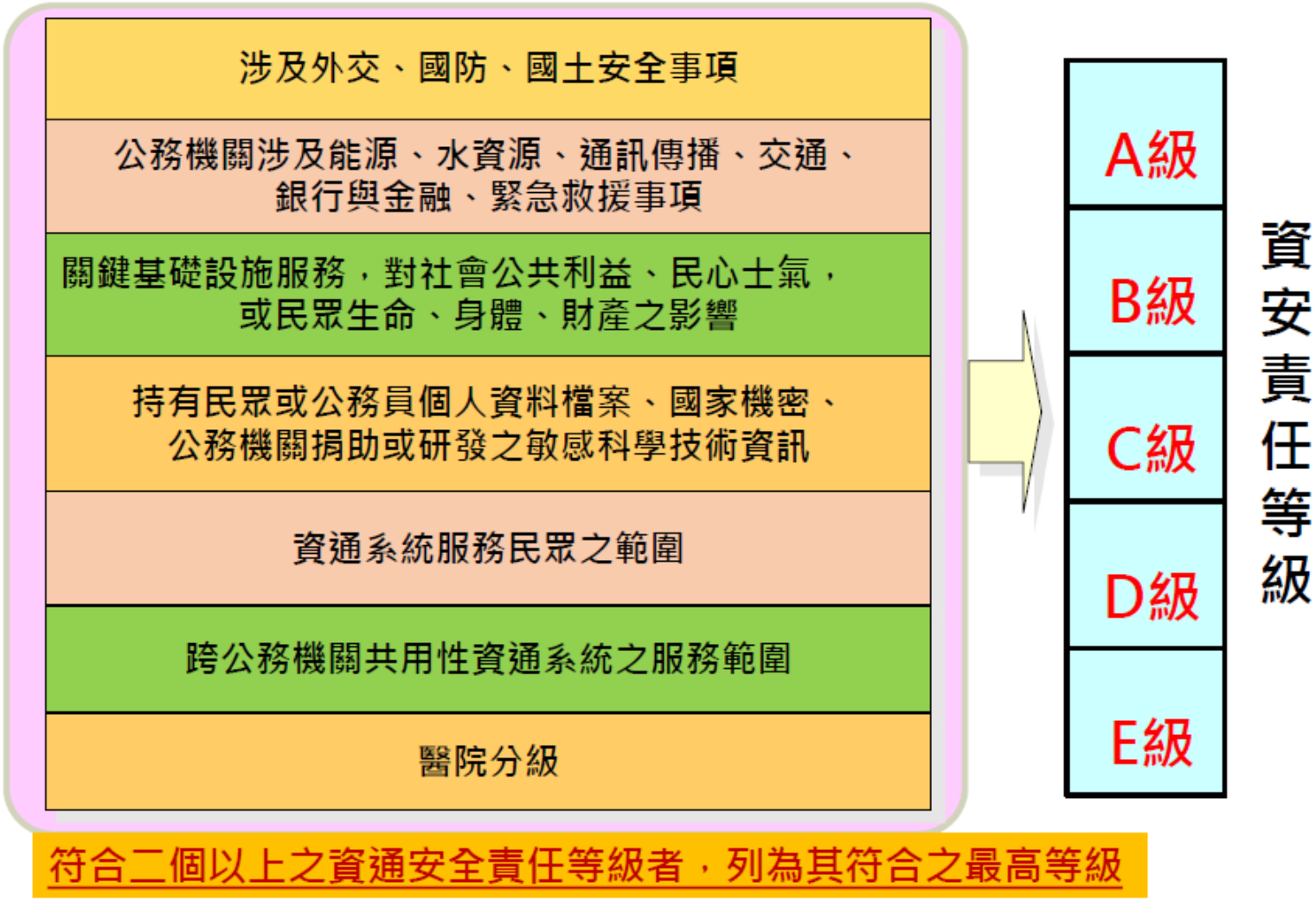
- 客製化開發者，應提供該資通系統之安全性檢測證明
- 非自行開發者，並應標示內容與其來源及提供授權證明。
- 受託者知悉資通安全事件時，應立即通知委託機關及採行之補救措施。
- 委託結束後，應確認受託者持有之資料之返還或刪除
- 受託者應採取之其他資通安全相關維護措施
- 委託機關應以稽核或適當方式確認受託者之執行情形

改善報告內容要求

- 稽核改善報告(§3)
  - 缺失或待改善之項目與內容
  - 發生原因
  - 所採取管理、技術、人力或資源等層面之措施
  - 預定完成時程及執行進度之追蹤

- 事件調查處理改善報告(§8)
  - 事件發生、完成損害控制或復原作業之時間
  - 事件影響之範圍及損害評估
  - 損害控制及復原作業之歷程、事件調查及處理作業之歷程
  - 事件根因分析
  - 防範再次發生所採取之管理、技術、人力或資源等層面之措施
  - 預定完成時程及成效追蹤機制

資安責任等級分級執行作業



應辦事項、資通系統防護分級

- 應辦事項：詳附表一至附表八
- 資通系統防護分級及防護基準：詳附件九及附表十
- 提報執行情形
  - 公務機關
  - 資通安全責任等級為A級或B級者，應依主管機關指定之方式，提報應辦事項之辦理情形
  - 特定非公務機關
  - 中央目的事業主管機關得要求所管，依指定之方式提報應辦事項之辦理情形

應辦事項-管理面

辦理項目	辦理內容	A	B	C
資通系統分級及防護基準	完成資通系統分級，並完成防護基準；每年至少檢視一次妥適性	1年內	1年內	2年內
資訊安全管理系統之導入及通過公正第三方之驗證	全部核心資通系統導入資訊安全管理系統，並於三年內完成第三方驗證；並持續維持其驗證有效性	2年內	2年內	2年內
業務持續運作演練	全部核心資通系統	每年1次	每2年1次	每2年1次
辦理內部資通安全稽核		每年2次	每年1次	每2年1次
資通安全專責人員(一年內)		專職(責)4人	專職(責)2人	專職(責)1人
資安治理成熟度評估(公務機關)		每年1次	每年1次	

應辦事項-技術面

辦理項目	辦理內容	A	B	C
安全性檢測	全部核心資通系統網站安全弱點檢測	每年2次	每年1次	每2年1次
	全部核心資通系統系統滲透測試	每年1次	每2年1次	每2年1次
資通安全健診	網路架構檢視、網路惡意活動檢視、使用者端電腦惡意活動檢視、伺服器主機惡意活動檢視、目錄伺服器設定及防火牆連線設定檢視	每年1次	每2年1次	每2年1次
資通安全威脅偵測管理機制	完成威脅偵測機制建置，並持續維運	1年內	1年內	
	依主管機關指定之方式提交監控管理資料(公務機關)	V	V	

應辦事項-技術面

辦理項目	辦理內容	A	B	C
資通安全防護(啟用，並持續使用及適時進行軟、硬體之必要更新或升級)	防毒軟體、網路防火牆、具有郵件伺服器者，應備電子郵件過濾機制	1年內	1年內	1年內
	IDS/IPS、具有對外服務之核心資通系統者，應備應用程式防火牆(WAF)	1年內	1年內	
	APT攻擊防禦	1年內		
政府組態基準	依主管機關公告之項目，完成政府組態基準導入作業，並持續維運(公務機關)	1年內	1年內	

應辦事項-認知與訓練

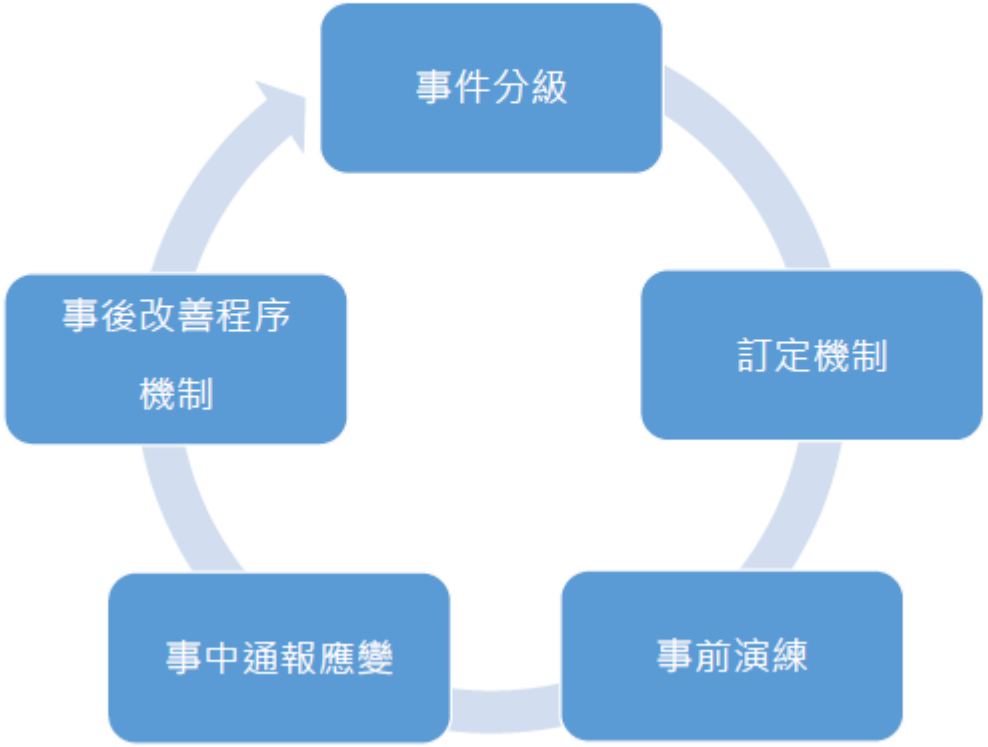
辦理項目	辦理內容	A	B	C
資通安全教育訓練	資通安全及資訊人員，每年接受之資通安全專業課程訓練或資通安全職能訓練	4名各12小時	2名各12小時	1名12小時
	一般使用者及主管，每人每年至少接受之一般資通安全教育訓練	3小時	3小時	3小時
資通安全專業證照及職能訓練證書	初次受核定或等級變更後之一年內，資通安全專職(責)人員總計應持有之資通安全專業證照，並持續維持證照之有效性	4張	2張	1張
	資通安全專職人員總計應持有之資通安全職能評量證書，並持續維持證照之有效性(公務機關)	4張	2張	1張

應辦事項-D級與E級

面向 作業 名稱 等級	技術面	認知與訓練
	資通安全防護	資通安全教育訓練
D級	初次受核定或等級變更後之一年內，完成下列資通安全防護措施之啟用，並持續使用及適時進行軟、硬體之必要更新或升級 一、防毒軟體 二、網路防火牆 三、具有郵件伺服器者，應備電子郵件過濾機制	一般使用者及主管，每人每年至少接受三小時以上之一般資通安全教育訓練
E級		一般使用者及主管，每人每年至少接受三小時以上之一般資通安全教育訓練

資通安全事件通報及應變辦法

- 為強化各機關之資安事件之因應。
- 規範事件之分級、事前演練、事中通報及應變，以及事後改善之程序、機制。



資通安全事件情資分享機制

情資分享

資通安全事件通報機制

其他情資

行政院建立資通安全情資分享機制

行政院、上級機關

中央目的事業主管機關

經濟部、交通部、金管會及通傳會等

公務機關資通安全事件通報(§13)(強制通報)

公務機關

非公務機關資通安全事件通報(§17)(強制通報)

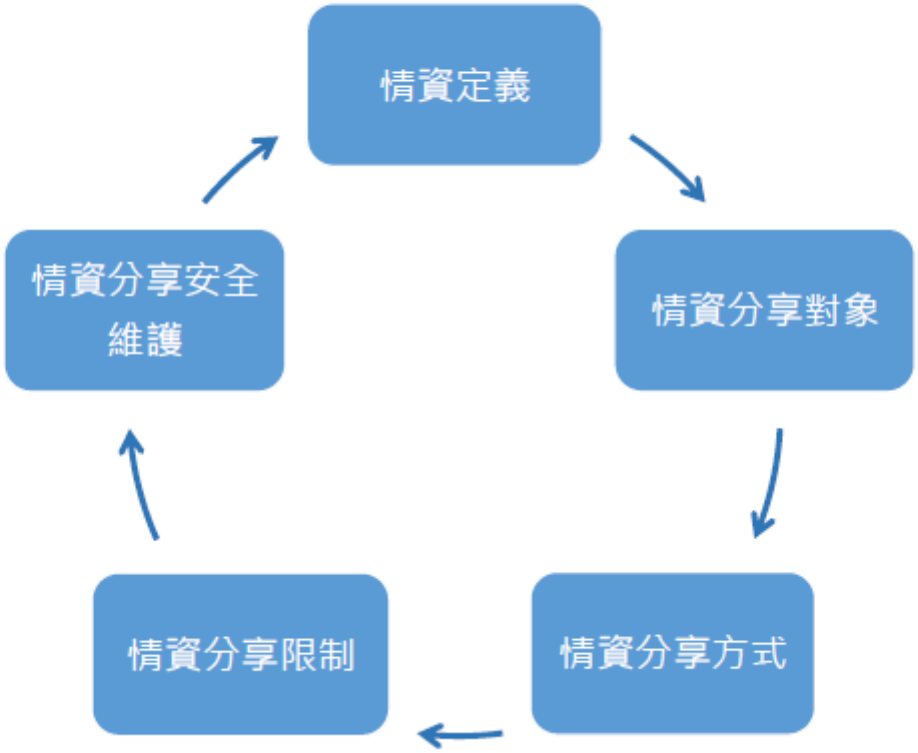
關鍵基礎設施提供者 ＋ 公營事業、政府捐助之財團法人

非公務機關資通安全事件通報 (自願通報)

所有非公務機關

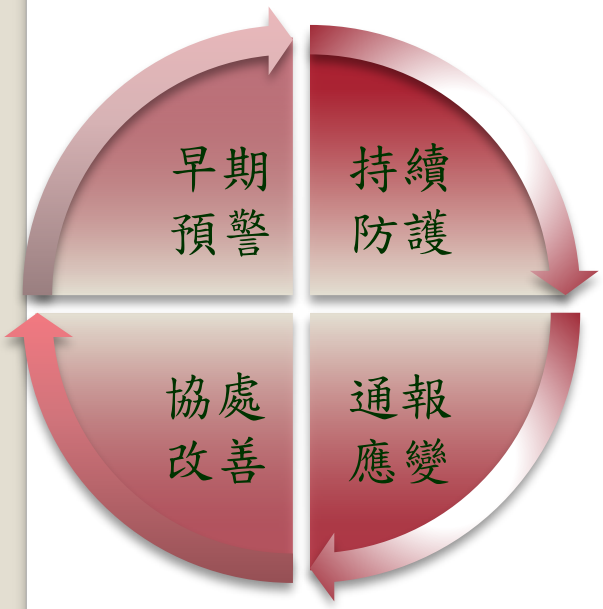
資通安全情資分享辦法

- 提升各機關對於資安之預警能力，強化資安相關資訊之交流。



資安聯防

使關鍵基礎設施**八大領域**均完成資安**四大面向**整備，建立情報驅動(Intelligence-based)之國家層級資安聯防架構



四大面向

X

八大領域

=

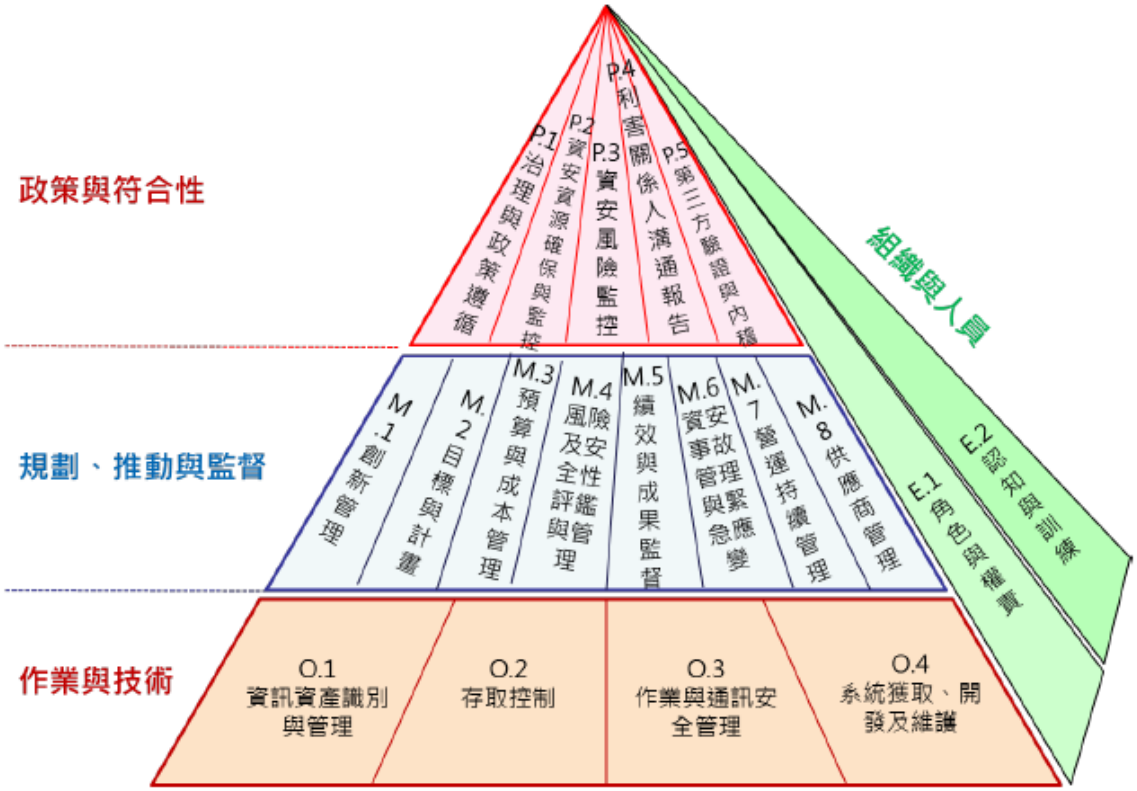
國家資安聯防架構

資安情報驅動三屬性

- 能見度(Visibility)
- 分析能量(Analysis)
- 快速行動(Action)

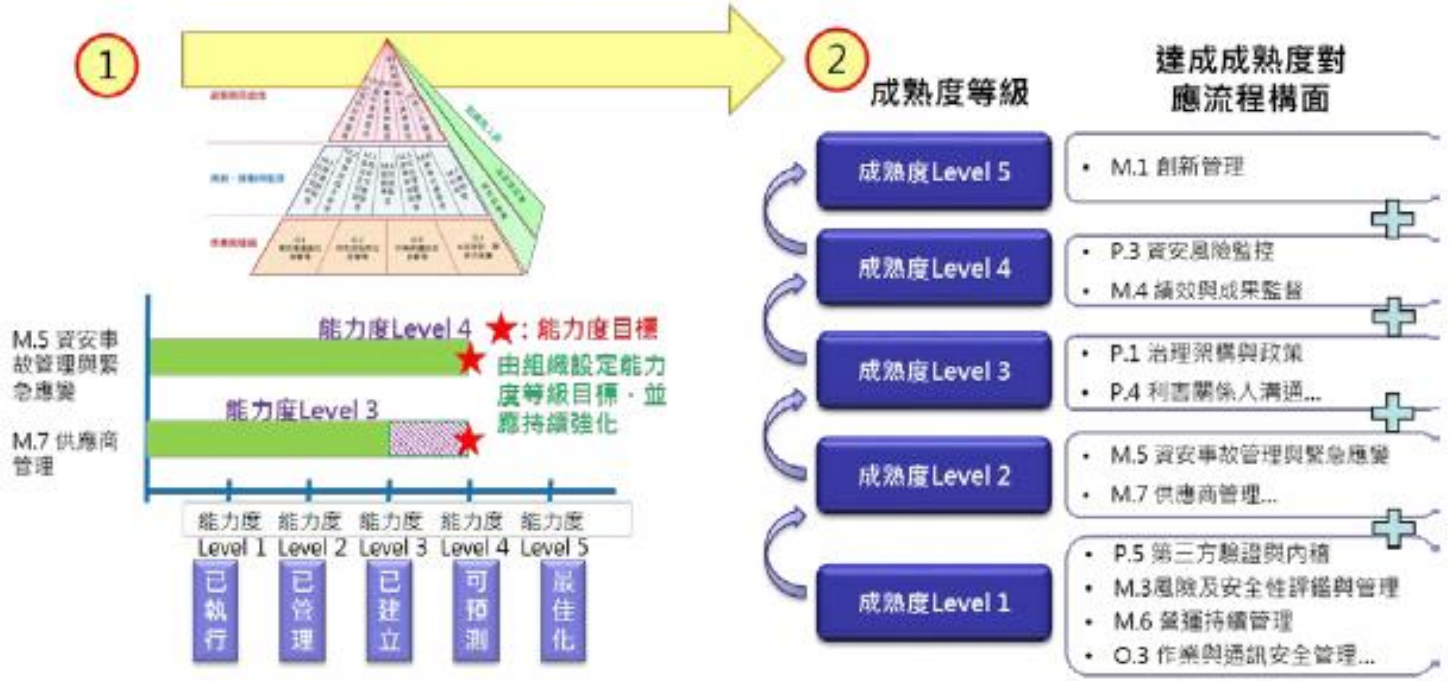
資安治理流程構面關係

依據資安治理架構模型之運作，各機關內的資安治理應與管理緊密配合並屬有關聯關係。以資安資源管理考量為例，在政策與符合性管理面向，資安治理強調組織整體之資安資源管理；在規劃、推動與監督面向

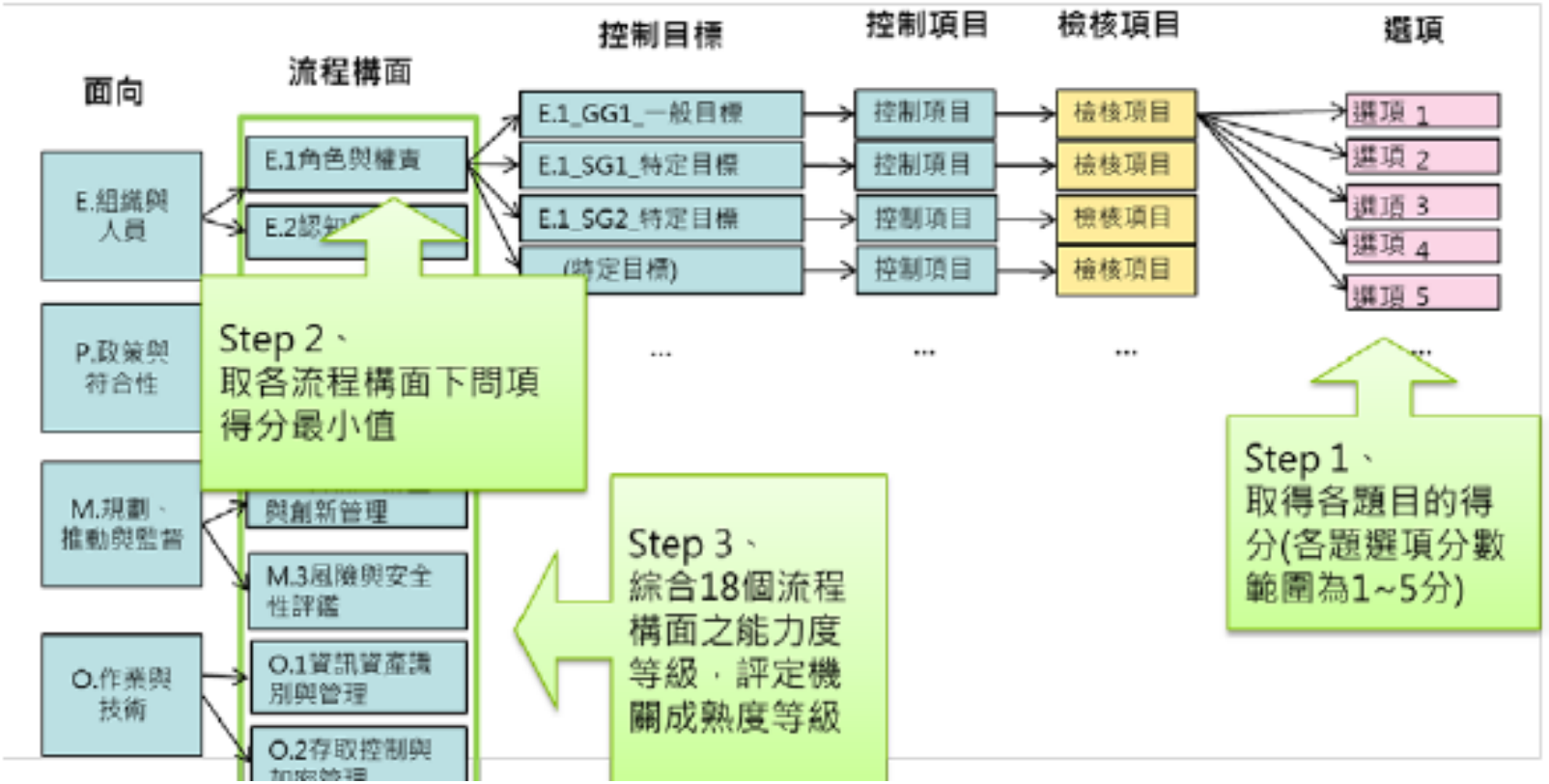


資安治理流程能力度與成熟度評審方法

- **能力度等級**：
  - 描繪組織流程於**特定**流程構面中的狀態
  - 以評審各流程構面之能力度
- **成熟度等級**：
  - 描繪組織的整體狀態
  - 用以評審組織之成熟度

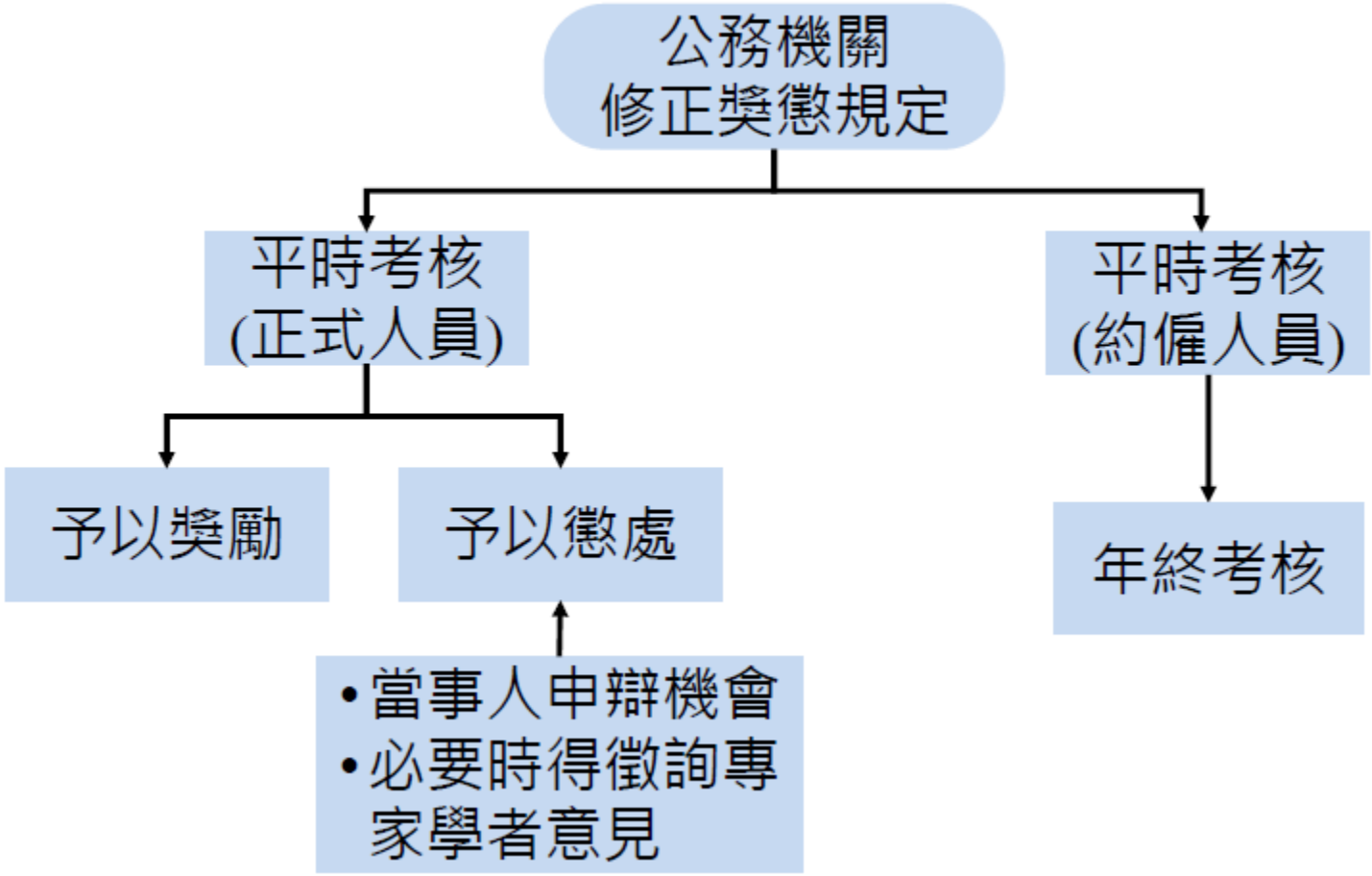


資安治理流程能力度與成熟度評審方法



公務機關所屬人員資通安全事項獎懲辦法

- 敦促公務機關所屬人員執行資通安全維護事務



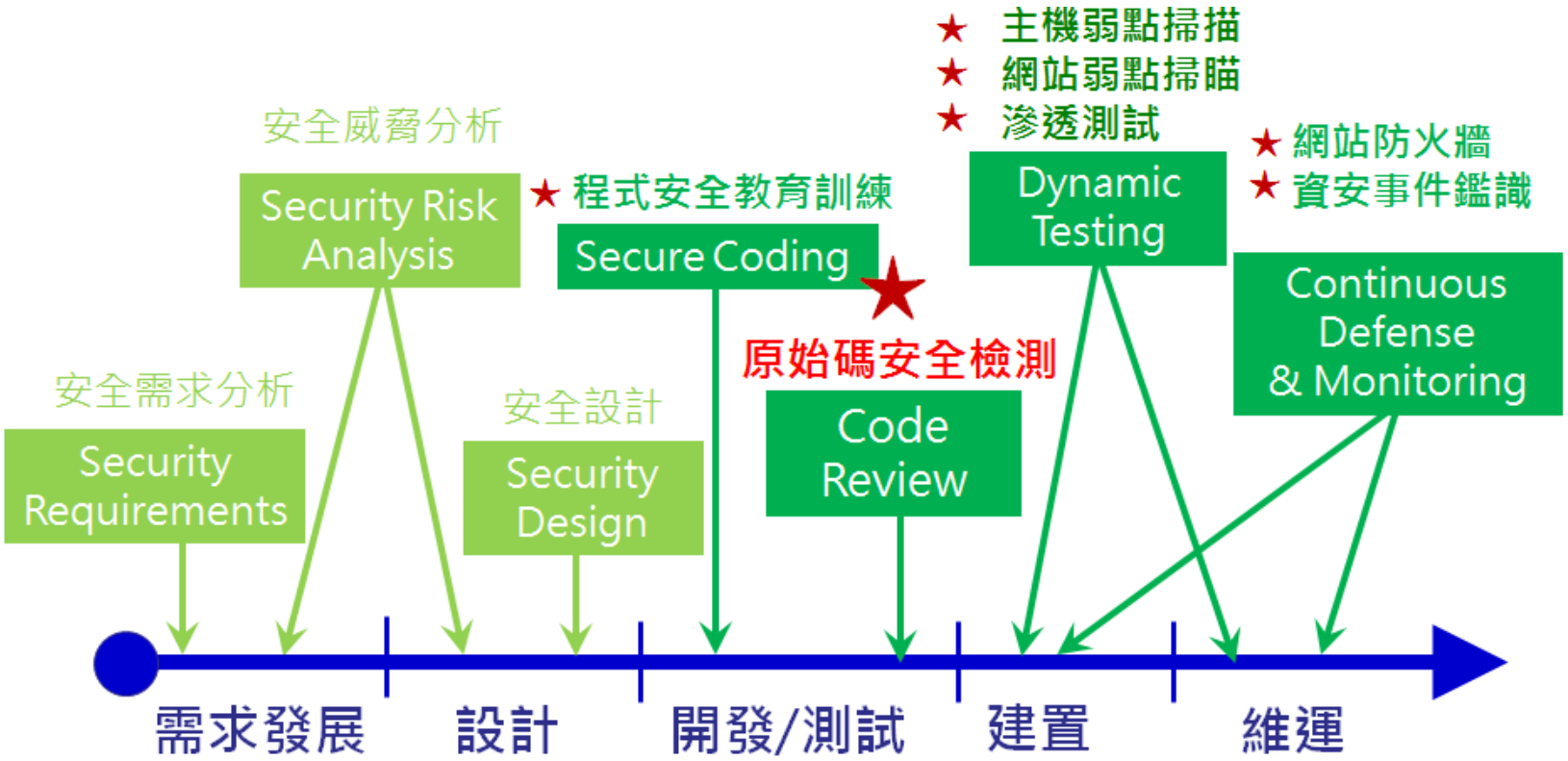
資通安全防護及控制措施

- **系統獲取、開發及維護**(有維護、自行或委外開發資通系統機關適用)
  - 資通系統應依「資通安全責任等級分級辦法」附表九之規定完成系統防護需求**分級**，依分級之結果，完成附表十中資通系統**防護基準**，並注意下列事項
    - 開發過程請依安全系統發展生命週期(Secure Software Development Life Cycle,SSDLC)納入資安要求，並參考行政院國家資通安全會報頒布之最新「安全軟體發展流程指引」、「安全軟體設計指引」及「安全軟體測試指引」
    - 開發前設計安全性要求，包含機敏資料存取、用戶登入資訊檢核及用戶輸入輸出之檢查過濾，並檢討執行情形
    - 上線前執行安全性要求測試，包含機敏資料存取、用戶登入資訊檢核及用戶輸入輸出之檢查過濾測試，並檢討執行情形
    - 執行資通系統源碼安全措施，包含源碼存取控制與版本控管，並檢討執行情形

Secure SDLC 方法論比較

	Cigital – TouchPoint Model	MS - SDL	OWASP - CLASP
1	• External Review	• 安全教育訓練	• 安全意識宣導方案
2	• Define Security Requirements • High Level Risk Assessments • Define Use & Misuse Cases	• 考量安全需求 • 進行風險評鑑 • 使用品質關卡	• 捕捉安全軟體需求 • 施行應用程式評鑑
3	• Secure Architecture & Design Patterns • Threat Modeling • Security Architecture Review • Security Test Planning • Technical Risk Assessment	• 減少攻擊面 • 威脅建模 • 檢視安全設計	
4	• Peer Code Review • Automated Code Review • Security Unit Tests	• 採用通過檢驗工具 • 源碼靜態分析 • 拒用不安全函式	• 實作安全開發實務
5	• White Box Testing • Black Box Testing	• 重新審視威脅模型與攻擊面 • 源碼動態分析	• 建置漏洞修補程序
6	• Secure Configuration • Secure Deployment • Incident Management • Vulnerability Management	• 最終安全審查 • 事故應變計畫	• 定義監測安全評量 • 發布作業安全指引

安全的軟體發展生命週期 (SSDLC)



委外開發規範

共通規範 - 國家資通安全研究院 (nat.gov.tw)

資訊作業委外資安參考指引
資訊服務採購案之資安檢核事項
資通系統委外開發RFP
資通系統資安需求項目查檢表
系統委外安全需求與驗證實務

軟體開發

共通規範 - 國家資通安全研究院 (nat.gov.tw)

資通系統獲取開發及維護程序書

106年WEB應用程式安全參考指引

附件 1 Web 應用程式安全查檢表

Web 應用程式安全檢核表						
控制措施	類別	實作項目	適用分級			是否符合
			普	中	高	
存取控制	帳號管理	使用者的會談階段，設定該帳號在合理的時間(至多 30 分鐘)內未活動即自動失效	◎	◎	◎	
		使用者的會談階段在登出後失效	◎	◎	◎	
		管理者介面限制存取來源或不允許遠端存取	◎	◎	◎	
	最小權限	對使用者/角色，僅賦予所需要的最低權限		◎	◎	
		軟體程序(process)及伺服器服務，以一般使用者權限執行，不以系統管理員或最高權限		◎	◎	
	遠端存取	採用伺服端的集中過濾機制檢查使用者授權	◎	◎	◎	
稽核與	稽核事件	針對身分鑑別失敗、存取資源失敗、重要行為、重要資料異動、功能錯誤及管理者行為進行日誌記錄	◎	◎	◎	

軟體開發

共通規範 - 國家資通安全研究院 (nat.gov.tw)

安全軟體測試參考指引v1.0_1031231.rar
安全軟體設計參考指引v1.0_1031031.rar
安全軟體發展流程指引v1.0_1030630.rar
源碼安全查檢表範例

項目	檢核結果		
資料驗證(Data Validation)	是	否	不適用
確認驗證資料合法性的機制存在	☐	☐	☐
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	☐	☐	☐
確認所有輸入資料的正確長度檢查存在	☐	☐	☐
確認輸入欄位、客戶端暫存、HTTP 協定標頭皆有被驗證	☐	☐	☐
確認資料的格式正確且包含的是合法的字元	☐	☐	☐
確認資料驗證機制存在於伺服器端	☐	☐	☐
檢查資料驗證是否確實發生	☐	☐	☐

資安規範

依照**資通安全責任等級分級辦法第 11 條**「各機關自行或委外開發之資通系統應依**附表九所定資通系統防護需求分級原則完成資通系統分級**，並依**附表十所定資通系統防護基準執行控制措...**」。
所以必須先藉由附表九，將委外開發的系統進行分級，並依照附表十的各等級應辦理之防護基準來落實資安防護，並將防護基準所要求的事項加入到委外的工作說明書。

資通安全管理法
資通安全責任等級分級辦法
附表一 資通安全責任等級A級之 公務機關 應辦事項
附表二 資通安全責任等級A級之 特定非公務機關 應辦事項制
附表九、資通系統防護需求分級原則
附表十、資通系統防護基準

資通系統籌獲各階段資安強化措施

「資通系統籌獲各階段資安強化措施」2022 年 5 月 26 日開始實施，試行一年。用以補充說明資通安全管理法施行細則第四條規定選任或監督受託者之相關行政流程及應注意事項。

各控制措施來源為資通安全責任等級分級辦法中的附表十《資通系統防護基準修正規定》，並參考 NCCST 的資通系統防護基準驗證實務

資通系統防護基準驗證實務(V1.1)_1110928.rar

資通系統防護基準類別

項次	構面	控制措施
1	存取控制	▪帳號管理 ▪最小權限 ▪遠端存取
2	事件日誌與可歸責性	▪記錄事件 ▪日誌紀錄內容 ▪日誌儲存容量 ▪日誌處理失效之回應 ▪時戳及校時 ▪日誌資訊之保護
3	營運持續計畫	▪系統備份 ▪系統備援
4	識別與鑑別	▪內部使用者之識別與鑑別 ▪身分驗證管理 ▪鑑別資訊回饋 ▪加密模組鑑別 ▪非內部使用者之識別與鑑別

資通系統防護基準類別

項次	構面	控制措施
5	系統與服務獲得	▪系統發展生命週期需求階段 ▪系統發展生命週期設計階段 ▪系統發展生命週期開發階段 ▪系統發展生命週期測試階段 ▪系統發展生命週期部署與維運階段 ▪系統發展生命週期委外階段 ▪獲得程序 ▪系統文件
6	系統與通訊保護	▪傳輸之機密性與完整性 ▪資料儲存之安全
7	系統與資訊完整性	▪漏洞修復 ▪資通系統監控 ▪軟體及資訊完整性

範例-某系統安全需求

- HTTPS傳輸加密
- 登入錯誤3次鎖定IP與帳號30分鐘
- 重設密碼功能使用圖形驗證碼(CAPTCHA)
- 網站下載資料提供HASH值供比對
- 系統程式與資料庫定時匯出備份至備援機
- 使用者輸入過濾特定SQL Injection惡意字元
- 密碼HASH過後儲存，不存純文字密碼
- 記錄使用者異動資料行為，保留原資料
- Log包含人事時地物
- 會談階段30分鐘失效
- 使用最新版函式庫，開啟port與服務最小化

軟體安全查檢表

OWASP - ASVS

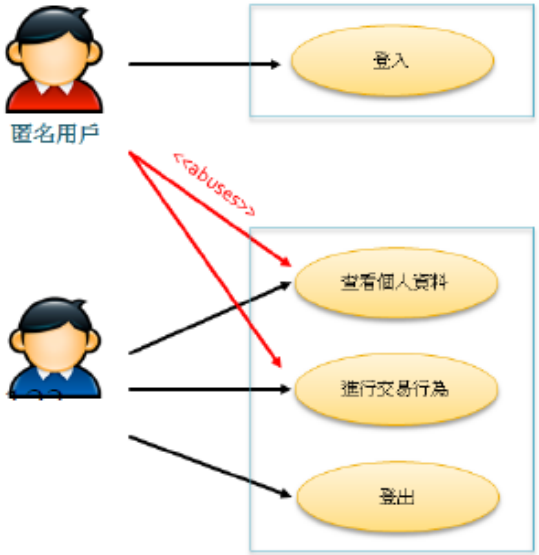
V2: Authentication Verification Requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

	AUTHENTICATION VERIFICATION REQUIREMENT	LEVELS 1	2	3
V2.1	Verify all pages and resources require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓
V2.2	Verify all password fields do not echo the user's password when it is entered.	✓	✓	✓
V2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓
V2.5	Verify all authentication controls (including libraries that call external authentication services) have a centralized implementation.			✓
V2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓
V2.7	Verify password entry fields allow or encourage the use of passphrases, and do not prevent long passphrases or highly complex passwords being entered, and provide a sufficient minimum strength to protect against the use of commonly chosen passwords.		✓	✓
V2.8	Verify all account identity authentication functions (such as registration, update profile, forgot username, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.		✓	✓
V2.9	Verify users can safely change their credentials using a mechanism that is at least as resistant to attack as the primary authentication mechanism.		✓	✓
V2.12	Verify that all authentication decisions are logged. This should include requests with missing required information, needed for security investigations.		✓	✓
V2.13	Verify that account passwords are salted using a salt that is unique to that account (e.g., internal user ID, account creation) and use bcrypt, scrypt or PBKDF2 before storing the password.		✓	✓

SANS – SWAT

誤用案例模型(Misuse Case)

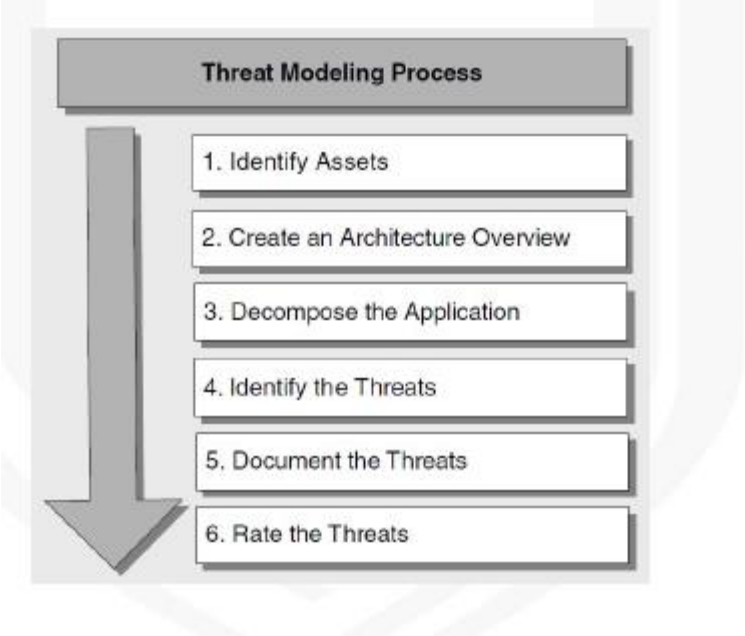


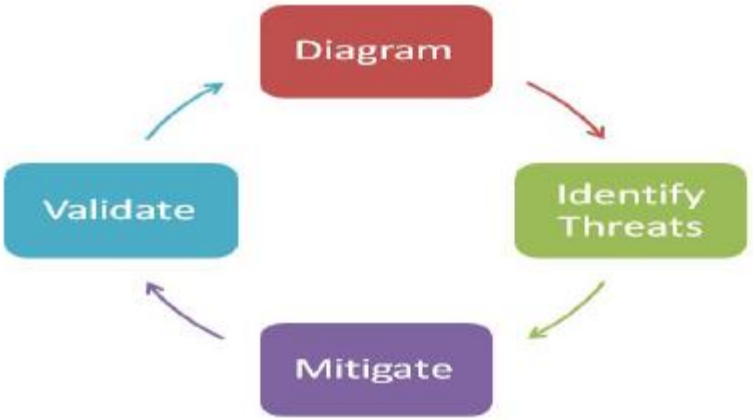
透過發展負面的使用情境來幫助識別安全需求
如何確保用戶只能存取查看個人資料?
- 每次存取要求都檢查其許可權
- 採用Server端的驗證授權機制，避免被繞過
- 避免直接顯示物件參考http://a.com/info?id=123

登入用戶的Session ID是否可能被劫奪進行偽冒交易
如何確保用戶只能存取查看個人資料?
- 會話識別字(Session ID)是隨機產生且不可預測
- 使用者的會話階段，設定在合理的時間內失效
- 使用者的會話識別字使用加密協定傳輸
- 使用者重新登入後，會話識別字(Session ID)會改變
- 不將會話識別字(Session ID)或使用者ID顯示於使用者可以改寫處

威脅建模(Threat Modeling)

- 威脅建模採用系統化的方法，以攻擊者角度，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瞭解，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





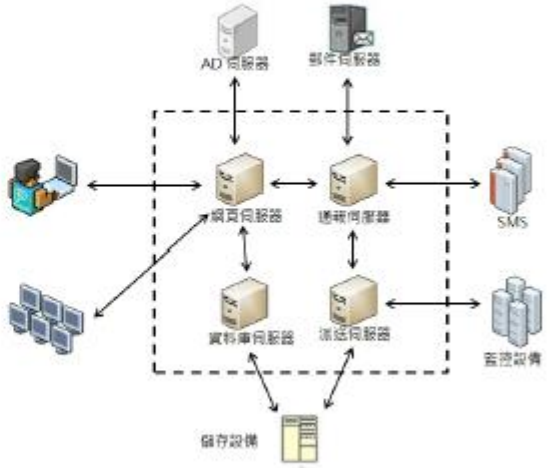
架構風險分析(Architecture Risk Analysis)

- 抗攻擊能力分析(Attack Resistance Analysis)
- 模糊分析(Ambiguity Analysis)
- 底層框架弱點分析(Underlying Framework Weakness Analysis)

抗攻擊能力分析 (Attack Resistance Analysis)

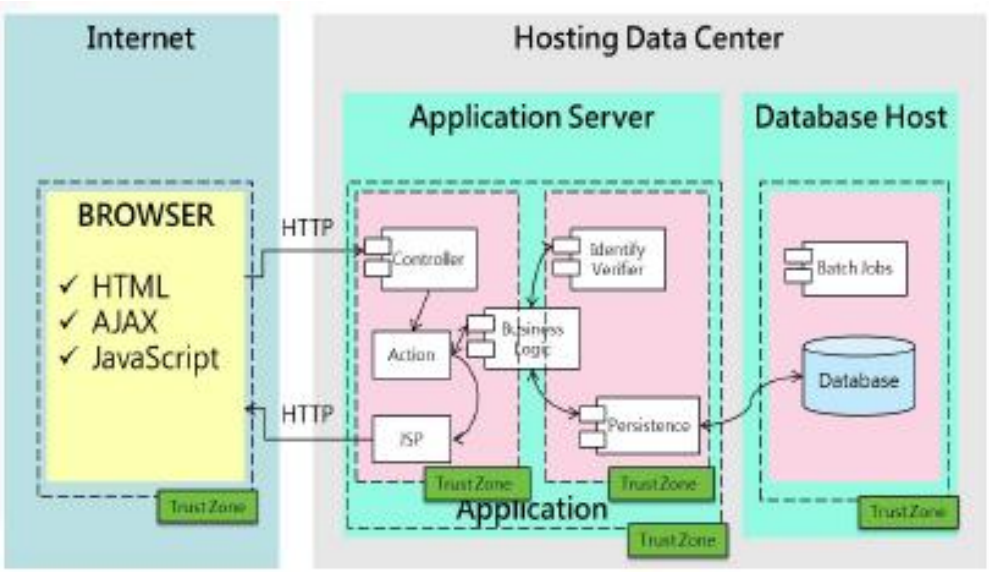
● 與「威脅建模」相似，但採用「已知」攻擊或弱點清單
– OWASP Top 10
– SANS Top 25
– WASC Attack & Weakness list

編號	攻擊
1	功能濫用(Abuse of Functionality)
2	暴力法攻擊(Brute Force)
3	緩衝區溢位(Buffer Overflow)
4	內容偽冒(Content Spoofing)
5	認證與會話神祕碼的預測(Credential/Session Prediction)
6	跨站腳本攻擊(Cross-Site Scripting, XSS)
7	跨站冒名請求(Cross-Site Request Forgery, CSRF)
8	阻絕服務(Denial of Service)
9	指紋探索與辨識(Fingerprinting)
10	格式化字串攻擊(Format String)
11	HTTP 回應偷渡(HTTP Response Smuggling)
12	HTTP 回應分割攻擊(HTTP Response Splitting)
13	HTTP 請求偷渡(HTTP Request Smuggling)
14	HTTP 請求分割攻擊(HTTP Request Splitting)
15	整數溢位(Integer Overflows)
16	LDAP 注入(LDAP Injection)
17	郵件命令注入(Mail Command Injection)
18	空字元注入(Null Byte Injection)
19	未經授權執行作業系統命令(OS Commanding)
20	路徑尋訪(Path Traversal)
21	可預測的資源位置(Predictable Resource Location)
22	遠端檔案含入(Remote File Inclusion, RFI)



模糊分析(Ambiguity Analysis)

- 用來發現新威脅的分析活動
- 需要兩組(位)以上對系統架構熟悉的人員
- 進行下列活動，然後比較其產出差異性，進行討論
  - 威脅建模
  - 敏感性資料建模



底層框架弱點分析 (Underlying Framework Weakness Analysis)

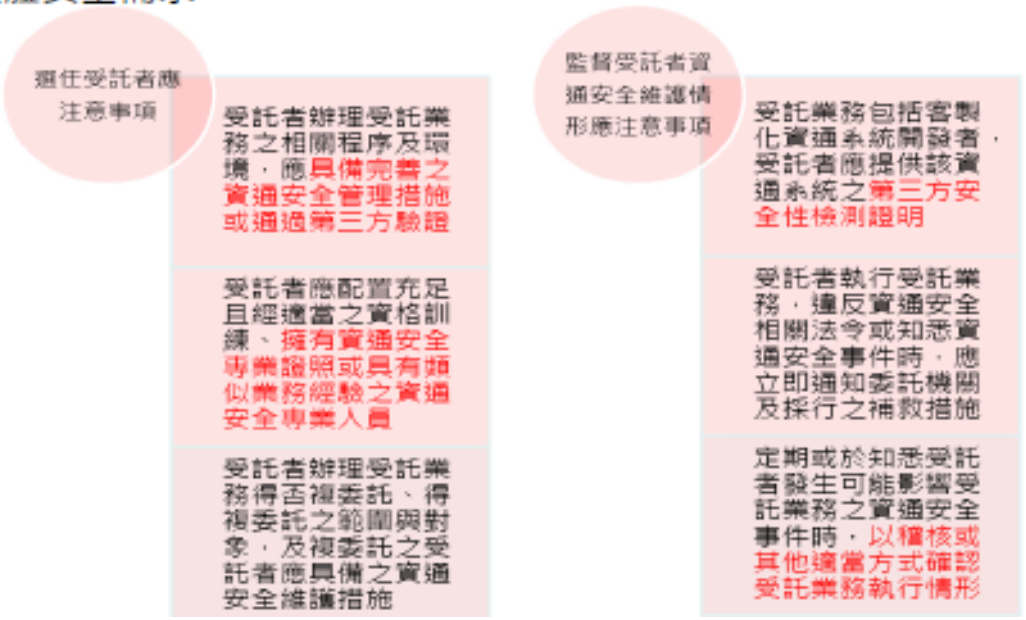
- 系統依賴其他底層軟體元件
- 底層軟體的安全問題影響系統安全
- 尋找底層軟體已知安全弱點

範例 –檢視底層軟體弱點

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
commons-validator-1.3.1.jar	cpe:/a:apache:apache_http_server:1.3.1	commons-validator:commons-validator:1.3.1	Medium	2	LOW	22
httpclient-4.2.3.jar	cpe:/a:apache:httpclient:4.2.3	org.apache.httpcomponents:httpclient:4.2.3	Medium	1	HIGHEST	14
mail.jar	cpe:/a:sun:javamail:1.4.2	javax.mail:mail:1.4.2	Medium	1	LOW	15
poi-3.6-20091214.jar	cpe:/a:apache:poi:3.6	org.apache.poi:poi:3.6	Medium	4	HIGHEST	12
solr-core-4.2.0.jar	cpe:/a:apache:solr:4.2.0		Medium	3	HIGHEST	8
spring-2.5.jar	cpe:/a:springsource:spring_framework:2.5.0 cpe:/a:vmware:springsource_spring_framework:2.5		High	7	HIGHEST	12
spring-mock-2.0-m4.jar	cpe:/a:springsource:spring_framework:2.0:m4 cpe:/a:vmware:springsource_spring_framework:2.0.m4	org.springframework:spring-mock:2.0-m4	High	6	HIGHEST	15
standard.jar	cpe:/a:apache:standard_taglibs:1.1.2		High	1	LOW	6
struts2-core-2.3.16.3.jar	cpe:/a:apache:struts:2.3.16.3	org.apache.struts:struts2-core:2.3.16.3	Medium	2	HIGHEST	13
struts2-tiles-plugin-2.3.16.3.jar	cpe:/a:apache:struts:2.3.16.3 cpe:/a:apache:tiles:2.3.16.3	org.apache.struts:struts2-tiles-plugin:2.3.16.3	Medium	2	HIGHEST	14
xwork-core-2.3.16.3.jar	cpe:/a:apache:struts:2.3.16.3	org.apache.struts.xwork:xwork-core:2.3.16.3	Medium	2	HIGHEST	23

資通系統或服務委外辦理之管理

- 依資安管理法施行細則第4條規定，委外辦理資通系統建置、維運或資通服務提供時，應考量受託者專業能力與經驗、委外項目之性質及資通安全需求，選任適當之受託者，並監督其資通安全維護情形
- 機關同仁辦理資訊作業委外時，可參考行政院國家資通安全會報頒布之最新「 政府資訊作業委外安全參考指引」，於資訊委外各階段，訂定具體安全需求



規劃分析

規劃分析

- 識別軟體中的資訊資產
  - 識別並整理該委外系統所包含的資訊資產以及安全目標
- 識別風險並產生安全需求
  - 考量法令法規或內部安全政策的需求
  - 識別資料等級
  - 系統安全需求項目查檢表
- 交付項目應包含安全測試結果
  - 系統弱點掃描(或滲透測試)結果
  - 系統源碼安全檢測及修復後複測結果紀錄

招標評選

招標評選

- 具有資安風險系統的相關開發經驗
- 廠商資安國際認證與人員資安專業證照數
  - Ex: CLSSP、ECSP、CISSP…etc.
- 具有軟體安全測試工具及問題修補能力
  - 程式碼安全檢測工具、滲透測試工具、弱點掃描工具…etc.

履約管理

履約管理

安全軟體測試階段

- 確認源碼檢測報告 中弱點已修正
- 確認滲透測試或弱　點掃描報告中弱點 已修正

安全軟體驗收階段

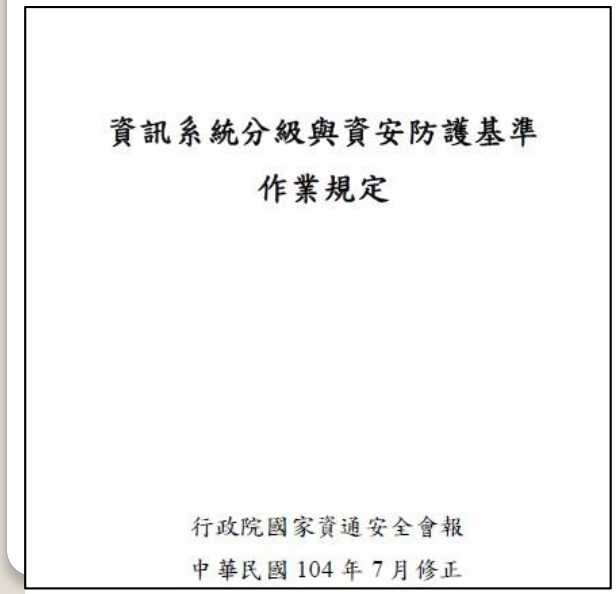
- 驗證系統安全需求
  項目實作之正確性

安全軟體部署與維運

- 確認強化部署環 境之程序
- 確認按照變更管　理程序執行軟體 維護

系統委外RFP之資安需求範本

- 依據104年教育訓練之回饋建議，提供政府機關「資訊系統委外開發RFP之資安需求範本」
- 參考OWASP、SANS等國際資安組織建議項目，並依據行政院資通安全會報頒布之「資訊系統分級與資安防護基準作業規定」，依資訊系統的安全分級(普、中、高)， 進行需求內容訂定與分級


資訊系統分級與資安防護基準
作業規定

行政院國家資通安全會報
中華民國 104 年 7 月修正

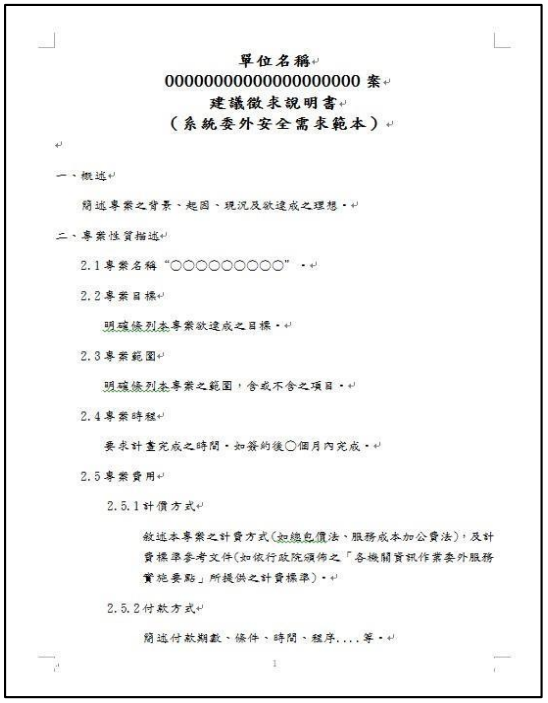

V2: Authentication Verification Requirements

The table below defines the corresponding verification requirements that apply for each of the verification levels. Verification requirements for Level 0 are not defined by this standard.

	AUTHENTICATION VERIFICATION REQUIREMENT	LEVELS 1	2	3
V2.1	Verify all pages and resources require authentication except those specifically intended to be public (Principle of complete mediation).	✓	✓	✓
V2.2	Verify all password fields do not echo the user's password when it is entered.	✓	✓	✓
V2.4	Verify all authentication controls are enforced on the server side.	✓	✓	✓
V2.5	Verify all authentication controls (including libraries that call external authentication services) have a centralized implementation.			✓
V2.6	Verify all authentication controls fail securely to ensure attackers cannot log in.	✓	✓	✓
V2.7	Verify password entry fields allow or encourage the use of passphrases, and do not prevent long passphrases or highly complex passwords being entered, and provide a sufficient minimum strength to protect against the use of commonly chosen passwords.		✓	✓
V2.8	Verify all account identity authentication functions (such as registration, update profile, forgot username, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.		✓	✓
V2.9	Verify users can safely change their credentials using a mechanism that is at least as resistant to attack as the primary authentication mechanism.		✓	✓
V2.12	Verify that all authentication decisions are logged. This should include requests with missing required information, needed for security investigations.		✓	✓
V2.13	Verify that account passwords are salted using a salt that is unique to that account (e.g., internal user ID, account creation) and use bcrypt, scrypt or PBKDF2 before storing the password.		✓	✓

RFP範本說明

- RFP範本之主體，以經濟部工業局公布之建議書徵求說明書(RFP)範例作為基礎，增加
  - 3.2系統安全需求
  - 附件1:系統安全需求項目查檢表



單位名稱
0000000000000000000 案
建議徵求說明書
（系統委外安全需求範本）

一、概述

簡述專案之背景、起因、現況及欲達成之理想。

二、專案性質描述

2.1 專案名稱“○○○○○○○○○”。

2.2 專案目標

明確條列本專案欲達成之目標。

2.3 專案範圍

明確條列本專案之範圍，含或不含之項目。

2.4 專案時程

要求計畫完成之時間，如簽約後○個月內完成。

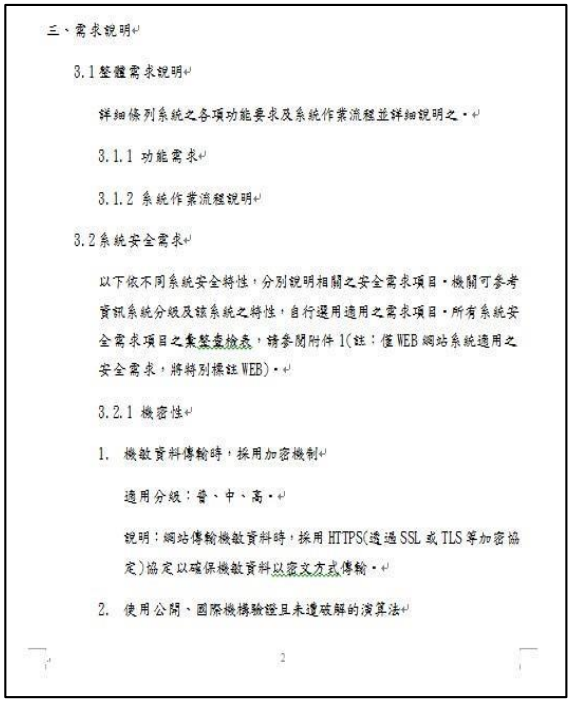
2.5 專案費用

2.5.1 計價方式

敘述本專案之計費方式(如總包價法、服務成本加公費法)，及計費標準參考文件(如依行政院頒佈之「各機關資訊作業委外服務實施要點」所提供之計費標準)。

2.5.2 付款方式

簡述付款期數、條件、時間、經序,...等。



三、需求說明

3.1 整體需求說明

詳細條列系統之各項功能要求及系統作業流程並詳細說明之。

3.1.1 功能需求

3.1.2 系統作業流程說明

3.2 系統安全需求

以下依不同系統安全特性，分別說明相關之安全需求項目，機關可參考資訊系統分級及該系統之特性，自行選用適用之需求項目，所有系統安全需求項目之彙整查檢表，請參閱附件 1(註：僅 WEB 網站系統適用之安全需求，將特別標註 WEB)。

3.2.1 機密性

1. 機敏資料傳輸時，採用加密機制

適用分級：普、中、高。

說明：網站傳輸機敏資料時，採用 HTTPS(透過 SSL 或 TLS 等加密協定)協定以確保機敏資料以密文方式傳輸。

2. 使用公開、國際機構驗證且未遭破解的演算法

採購需求資訊安全條約參考

資訊安全

一 安全檢測

1. 網站弱點掃描-廠商應依專案期程規定，全系統須通過交付經第三方網站弱點掃瞄工具之檢測，並提供檢測報告，相關檢測工具由得標廠商負責提供。弱點掃瞄工具必需包含OWASP Top 10最新，且檢測結果不得有高、中、低等級之風險。
2. 程式碼檢測-當交付未加密程式碼時，必須同時交付經第三方工具檢測此程式碼完整的安全漏洞檢測報告，且檢測結果不得有高、中、低等級之風險。
3. 系統壓力測試-驗證方式為本機關網路環境及本案硬體同體規格伺服器，模擬（300）個使用者同時上線，以（15）分鐘為測試期間，重複存取網站首頁，(95%)之要求其回應時間應達（3）秒以內的服務水準。以上須提供測試報告，內容至少需包括測試軟體名稱、測試人員、測試時間、測試方式、測試參數、測試環境軟硬體資訊、完成及回應時間等。（要求之數值由各系統訂定）

採購需求資訊安全條約參考

資訊安全

二、系統防護(核心系統需納入規範，符合下列要求為原則或提出經本中心同意之方替代機制後實施；非核心系統納入可提高系統之安全性)

1.使用者登入後的第一個非公開存取頁面，必須已通過身分驗證才允許存取
2.當閒置時間或可使用期限時間達(十)分鐘，系統應自動將使用者登出。
3.已確實規範使用者密碼強度，且符合密碼長度12個字元以上、組成須包含英文大寫、英文小寫、數字以及特殊字元，4種字元至少包含3種
4.使用者更換密碼時，至少不可以與前3次使用過之密碼相同
5.具備帳戶鎖定機制，帳號登入進行身分驗證失敗達5次後，至少15分鐘內不允許該帳號繼續嘗試登入
6.採用圖形驗證碼(CAPTCHA)機制於身分驗證機制，如登入頁面或密碼更換頁面，以防範自動化程式之嘗試
7.密碼重設機制對使用者重新身分確認後，發送一次性及具有時效性令牌(Token)，例如:簡訊驗證碼、EMAIL驗證連結，檢查傳回令牌有效性後，才允許使用者進行重設密碼動作
8.發生錯誤時，頁面不會出現詳細的程式除錯訊息
9.HTTPS加密連線機制僅採用TLS1.1(含)以上版本
10.HTTPS連線未使用遭破解加密演算法，包括DES、3DES、RC4

採購需求資訊安全條約參考

資訊安全

三、保全開發環境

1.非經本機關同意，得標廠商嚴禁於本機關實際環境進行軟體開發或測試。

2.得標廠商測試時優先使用模擬資料，若因特殊事由使用真實資料進行測試時，亦不得使用原始資料，並對於使用之資料善盡保管及保密之責任。系統應對於異常事件、重要事件或特殊權限帳號存長保留日誌紀錄。

3.得標廠商對軟體之異動，應事先作好資料備份工作，並就其實施細節及可能之風險完成規劃及評估，經本機關確認方可實施，如有意外狀況發生時，除應採取還原或其他措施避免或減少不良影響，並於第一時間通知本機關。

4.作業系統、資料庫及應用程式之所有密碼資料，皆不得以明碼型態存放或傳輸，若有特殊需求須經本機關同意後辦理。

5.於開發或進行原程式修改之原始碼，須提供未加密完整原始碼且至少保留三代，交由本機關保管作為系統維護之用，系統相關軟體如有修改時應配合一併更新。

採購需求資訊安全條約參考

資訊安全

四、資安責任：

- 得標廠商應遵守行政院所頒訂之各項資訊安全規範及標準，並遵守資訊安全管理及保密相關規定。本機關保有對得標廠商執行稽核的權利。
- 得標廠商交付之軟硬體及文，應先行檢查是否內藏惡意程式（如病毒、蠕蟲、特洛伊木馬、間諜軟體等），並於上線前應清除環境之測試資料與帳號及管理資料與帳號。
- 本系統需依「政府機關 (構)資通安全等級分級作業規定」，需依資訊系統(「高」)等級，執行相關防護基準。(依各系統等級填列)
- 得標廠商需簽訂資訊保密切結書，並遵守本機關「資訊安全管理要點」相關規定。
- 本機關已導入資訊安全管理系統規範，驗證期間如需得標廠商配合辦理之事項，得標廠商應配合之，如有生相關費用，得標廠商應自行吸收。
- 本機關不定期進行安全檢測，如發現存在弱點，得標廠商應於一週內完成中高風險弱點修補，一個月內完成所有弱點修補（除系統誤判外），惟若修補事宜涉及原廠、第三方之情形，經本機關同意可排除。
- 契約履約或終止後，得標廠商應刪除或銷毀執行服務所持有機關之相關資料，或依機關之指示返還之，並保留執行紀錄。
- 本契約因期限屆滿、解除或其他原因而終止時，得標廠商及其工作人員仍負有前款之保密責任。

採購需求資訊安全條約參考

資訊安全

四、資安責任：

- 得標廠商在本案系統開發過程中、交付之產品及相關軟體項目如包含第三人開發之產品，應切結保證並提供授權證明文件，以證明軟體使用之合法性(以符合中華民國著作權法規範為準)，並提供手冊、磁片或光碟片，若發生侵害第三人合法權益時，由得標廠商負責處理，並承擔一切法律責任。得標廠商如有隱瞞事實或使用未授權軟體之行為，致使本機關遭致任何損失或聲譽之損害時，得標廠商應負一切損失賠-償與責任，並放棄法律之先訴抗辯權，且維持本案系統之正常運作。
- 得標廠商如需使用本機關之電腦設備、網路連線及辦公處所時，應依本機關作業規定辦理，並應盡善良管理人之責任，同時以在正常上班時間內使用為原則，如須逾時使用，得標廠商應另行提出申請。
- 得標廠商程式撰寫符合W3C 規範，並依循「Web 應用程式安全參考指引」(http://www.icst.org.tw/CommonSpecification.aspx?lang=zh)。
- 得標廠商提供服務，如發生資安事件時，必須通報本機關，提出緊急應變處置，並配合本機關做後續處理。
- 本專案所使用主機之作業系統或相關軟體工具發現安全漏洞時，得標廠商必須無條件進行修補並於2週內研擬修補之評估報告(必要時需配合修改所開發之程式，及更新所提供之系統工具版本)，經本機關同意後依規定時程執行相關修補作業。
- 廠商所提供之服務，如為軟體或系統發展，須針對各版本進行版本管理，並依照資安管理相關規範提供權限控管與存取紀錄保存。

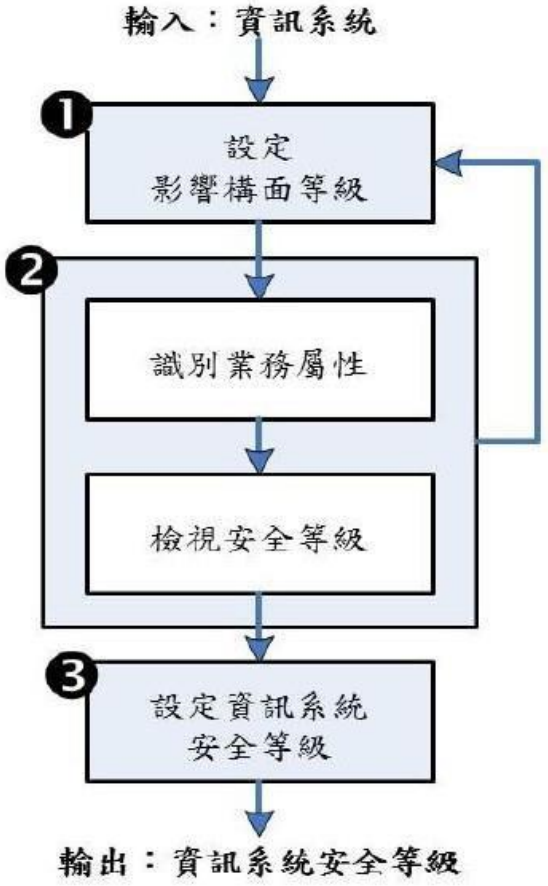
針對安全特性類型訂定

共50項安全需求項目

機密性 (Confidentiality)	完整性 (Integrity)	可用性 (Availability)
6項	6項	2項
身分驗證 (Authentication)	授權 (Authorization)	稽核 (Auditing)
12項	7項	4項
會談管理 (Session Management)	錯誤及例外處理 (Error and Exception Handling)	組態管理 (Configuration Management)
5項	3項	5項

依資訊系統安全分級

- 普級
  - –11項
- 中級
  - –11項+17項=28項
- 高級
  - –11項+17項+22項=50項



適用於全部等級的11項目

項次	安全特性	項目
1	機密性	機敏資料傳輸時，採用加密機制
2	完整性	於伺服器端以正規表示式(Regular Expression)方式，檢查使用 者輸入資料合法性
3	可用性	重要資料定時同步至備份或備援環境，並加以保護限制存取
4	身分驗證	除了允許匿名存取的功能外，所有功能都必須已通過身分驗證才允許存取
5	授權	執行功能及存取資源前，檢查使用者授權
6	日誌紀錄	針對身分驗證失敗、存取資源失敗、重要行為、重要資料異動、 功能錯誤及管理者行為進行日誌記錄

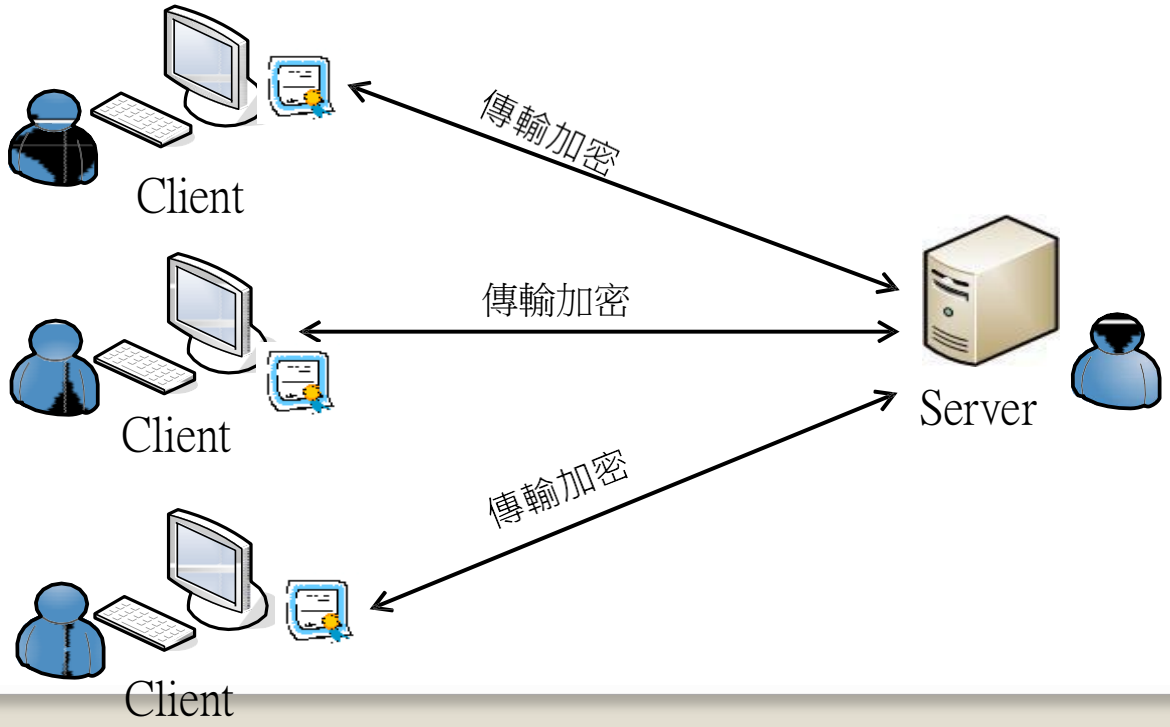
適用於全部等級的11項目

項次	安全特性	項目
7	日誌紀錄	日誌紀錄包含以下項目1.識別使用者之ID(不可為個資類型)。2.經系統校時後的時間戳記。3.執行的功能或存取的資源。4.事件類型或等級(priority)。5.事件描述
8	會談管理	使用者的會談階段，設定該帳號在合理的時間(至多30分鐘)內未活動即自動失效
9	會談管理	使用者的會談階段在登出後失效
10	錯誤及例外處理	發生錯誤時，使用者頁面僅顯示簡短錯誤訊息及代碼，不包含 詳細的錯誤訊息
11	組態管理	管理者介面限制存取來源或不允許遠端存取

1.傳輸加密 (1/3)

機密性

- 需求項目
  - 機敏資料傳輸時，採用加密機制
- 說明
  - 系統傳輸機敏資料時，採用SSL、TLS等加密協定， 以確保機敏資料以密文方式傳輸。



1.傳輸加密 (2/3)

機密性

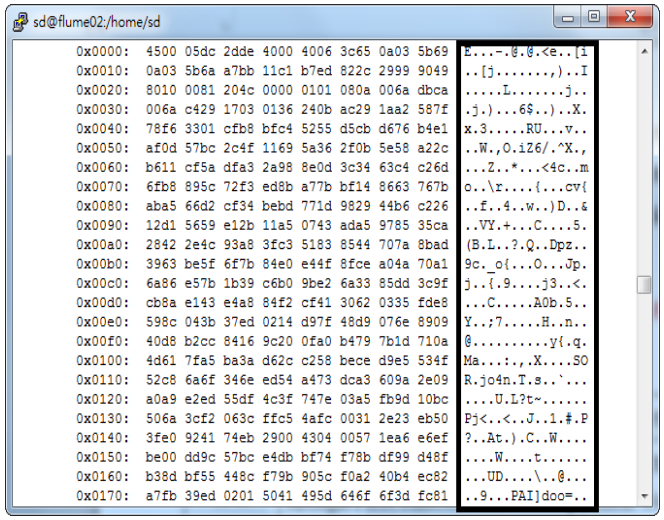
- 實作方法
  - 產生金鑰並申請伺服器SSL憑證
  - 調整伺服器設定檔

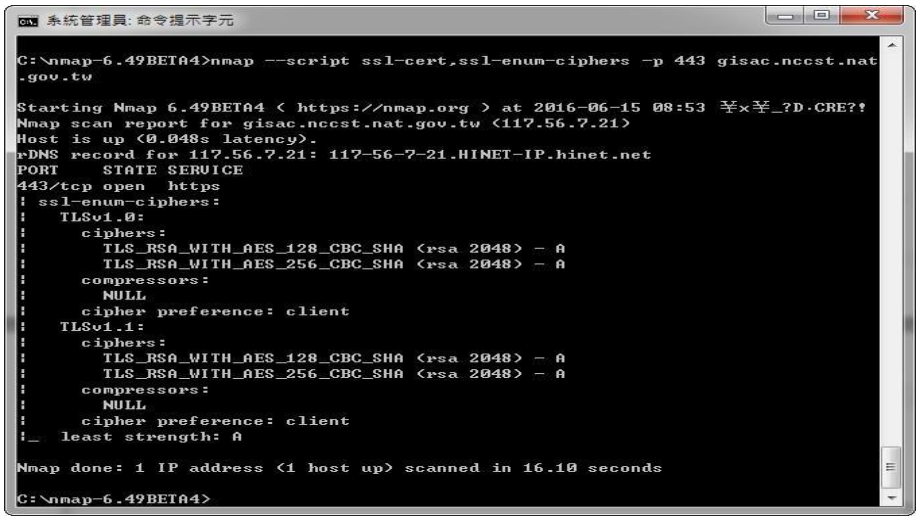
```
<Connector
    protocol="org.apache.coyote.http11.Http11NioProtocol"
    port="443" maxThreads="200“
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="%HOME%/.keystore" keystorePass="changeit"
    sslProtocol="TLSv1.2" sslEnabledProtocols="TLSv1.1,TLSv1.2"
    clientAuth="false“
    ciphers="TLS_RSA_WITH_AES_256_CBC_SHA256,
                TLS_RSA_WITH_AES_256_CBC_SHA,
                TLS_RSA_WITH_AES_128_CBC_SHA256,
                TLS_RSA_WITH_AES_128_CBC_SHA"
/>
```

1.傳輸加密 (3/3)

機密性

- 驗證方法
  - 瀏覽器查看網站憑證
  - 工具驗證加密協定
  - 網路封包側錄分析





2.輸入驗證(1/3)

完整性

- 需求項目
  - 於伺服器端以正規表示式(Regular Expression)方式，檢查使用者輸入資料合法性

- 說明
  - 必須定義何謂合法的使用者輸入
    - ➢數字　➢符號
    - ➢英文　➢中文
    - ➢合法長度(Min, Max)
    - ➢特定格式(Email、電話、網址、密碼等，有其特殊規則)

2.輸入驗證(2/3)

完整性

- 實作方法
  - 採用程式語言之正規表示式API
  - Java為java.util.regex
  - NET為System.Text.RegularExpressions
  - PHP為Preg相關函式

```
public static boolean check(String password){
  boolean pass = false;
  String regex = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])"
      + "(?=.*[~`!@#$%^*()_+={}|\\[\\]\\\\])"
      + "[0-9a-zA-Z~`!@#$%^*()_+={}|\\[\\]\\\\]{12,}";
  Pattern inputPattern = Pattern.compile(regex);
  Matcher m = inputPattern.matcher(password);
  pass = m.matches();
  return pass;
}
```

2. 輸入驗證(3/3)

完整性

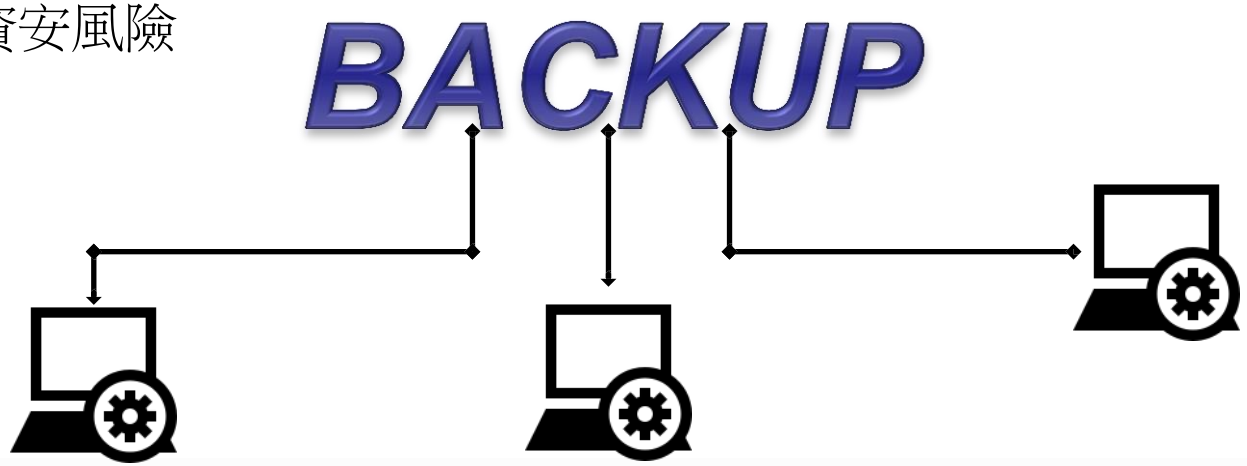
- 驗證方法
  - 須編製測試案例，以符合規則與不符合規則的資料，對輸入欄位進行測試
  - 以密碼複雜度(12碼以上，英文大小寫、數字、特殊符號皆最少一個)為例
    - 長度12碼以上，含英文大小寫及數字，不含特殊字元
    - 長度12碼以上，含英文大寫及數字，含特殊字元，不含英文小寫
    - 長度12碼以上，含英文小寫及數字，含特殊字元，不含英文大寫
    - 長度12碼以上，含英文大小寫，含特殊字元，不含數字
    - 長度小於12碼，含英文大小寫及數字，含特殊字元

3. 定時備份(1/2)

可用性

- 需求項目
  - 重要資料定時同步至備份或備援環境，並加以保護限制存取

- 說明
  - 系統具備重要資料定時備份機制，依組織規範將資料同步至備份或備援環境，以避免系統毀損或資料綁架勒索對資料可用性之危害。重要資料於備份或備援環境應有保護限制存取措施，以避免增加其他資安風險



3. 定時備份(2/2)

可用性

- 實作方法
  - 網站資料庫進行定時匯出，加密傳送遠端進行備份。
  - 亦可採用DB本身的同步機制
- 驗證方法
  - 進行備援演練及還原測試
- MySQL 匯出與回存範例
  - mysqldump -u [user] -p[password] --all-databases > backup.sql
  - mysqlimport -u [user] -p[password] [DB] backup.sql

4. 身分驗證(1/2)

身分驗證

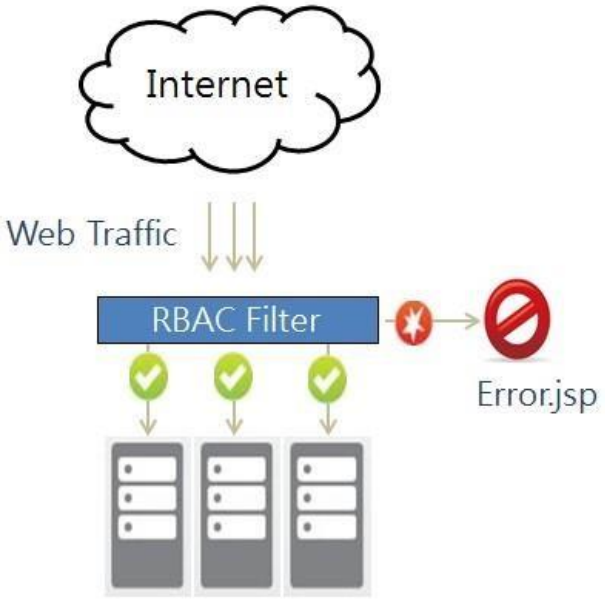
- 需求項目
  - 除了允許匿名存取的功能外，所有功能都必須已通過身分驗證才允許存取

- 說明
  - 網站除公開區域外，其他網頁皆需已進行身分驗證登入成功後，才得以存取。
  - 使用者若存取非公開區域，檢查機制發現其尚未通過身分驗證時，應不允許其存取頁面並將其導向至首頁或登入頁面。

4. 身分驗證(2/2)

身分驗證

- 實作方法
  - 採用一致全面性，強制適用於全系統的授權及存取控制機制
  - 網站建議採用過濾器(Filter)機制，強制設定管制範圍
- 驗證方法
  - 以測試案例驗證，當尚未通過身分驗證時，存取管制範圍內的網頁或功能，會被拒絕存取

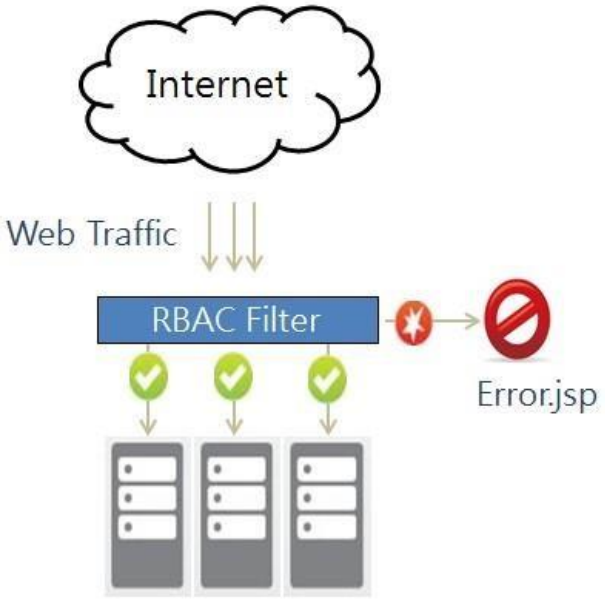



```
<filter>
  <filter-name>AccCtrlFilter</filter-name>
  <filter-class>sweb.ch7.AccCtrlFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>AccCtrlFilter</filter-name>
  <url-pattern>/private/*</url-pattern>
</filter-mapping>
```

4. 身分驗證(2/2)

身分驗證

- 實作方法
  - 採用一致全面性，強制適用於全系統的授權及存取控制機制
  - 網站建議採用過濾器(Filter)機制，強制設定管制範圍
- 驗證方法
  - 以測試案例驗證，當尚未通過身分驗證時，存取管制範圍內的網頁或功能，會被拒絕存取




```
<filter>
  <filter-name>AccCtrlFilter</filter-name>
  <filter-class>sweb.ch7.AccCtrlFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>AccCtrlFilter</filter-name>
  <url-pattern>/private/*</url-pattern>
</filter-mapping>
```

5. 檢查授權(1/3)

授權

- 需求
  - 執行功能及存取資源前，檢查使用者授權

- 說明
  - 系統中除了公開區域外，任何執行功能及存取資源動作前，應檢查使用者已通過身分驗證且使用者具備權限可執行該功能或存取該項資源

5. 檢查授權(2/3)

授權

- 實作方法
  - 採用過濾器(Filter)機制， 先檢查是否通過身分驗證，再檢查是否具有存取網頁或功能的權限

```
public class AccCtrlFilter implements Filter {
  public void doFilter(ServletRequest request, ServletResponse response,
      FilterChain chain) throws IOException, ServletException {
    HttpServletRequest req = (HttpServletRequest) request;
    HttpServletResponse res = (HttpServletResponse) response;
    HttpSession session = req.getSession();
    req.setCharacterEncoding("UTF-8");
    res.setCharacterEncoding("UTF-8");
    String user_id = (String) session.getAttribute("id");
    Map priv = (Map) session.getAttribute("priv");
    if (user_id == null || priv == null) {
      res.sendRedirect(req.getContextPath()+"/Login.jsp");
      return;
    }
    String uri = req.getRequestURI();
    if (priv.containsKey(uri)) {
      chain.doFilter(request, response);
    }else{
      res.sendRedirect(req.getContextPath()+"/Error.jsp");
      return;
    }
  }
```

未通過驗證
重導至首頁

不具權限
重導至錯誤
訊息頁

5. 檢查授權(3/3)

授權

- 驗證方法
  - 以測試案例驗證，不具有特定網頁或功能之權限，但嘗試存取時，會被拒絕存取
  - 以測試帳號A、B交互驗證是否可存取他人資源

6. 行為紀錄(1/3)

稽核

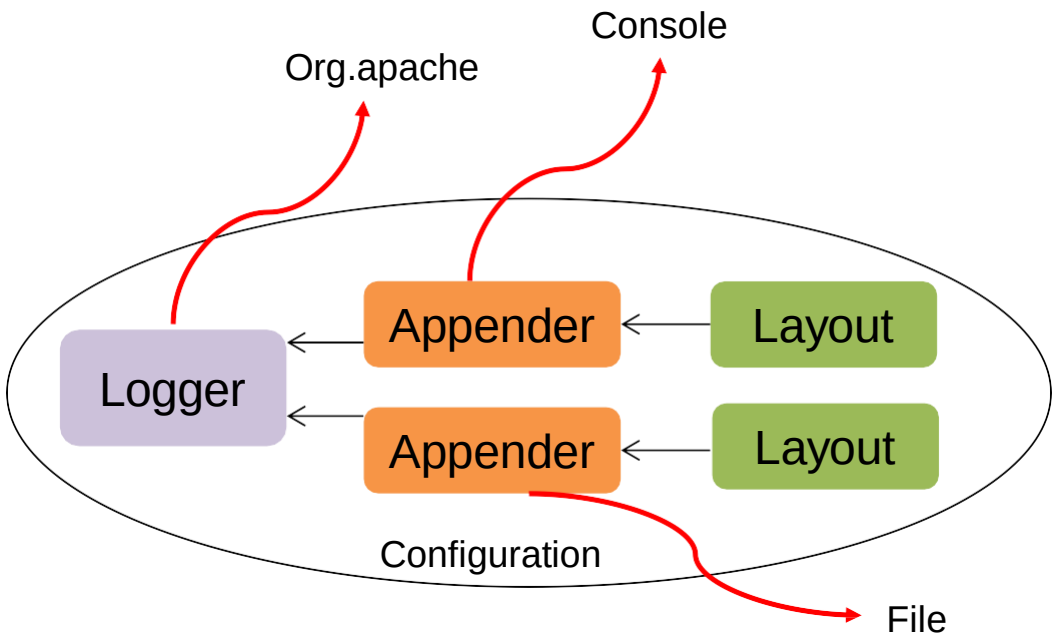
- 需求項目
  - 針對身分驗證失敗、存取資源失敗、重要行為、重要資料異動、功能錯誤及管理者行為進行日誌記錄

- 說明
  - 系統留存日誌紀錄之目的包含程式除錯、行為歸責、稽核取證及法規要求等。記錄身分驗證失敗、存取資源失敗、重要行為、重要資料異動、功能錯誤及管理者行為等，將有助於定期稽核系統行為及資安事件追查

6. 行為紀錄(2/3)

稽核

- 實作方法
  - 建議系統採用一致性的日誌紀錄(Log)機制，例如Log4J，以供系統不同功能直接套用
  - 透過設定檔控制
    - 程式碼範圍
    - 紀錄層級(Log Level)
    - 輸出目標
    - 輸出格式



6. 行為紀錄(3/3)

稽核

- 驗證方法
  - 就上述行為，以測試案例實際進行測試
  - 確認測試動作完成後，日誌紀錄正常產生且格式正確

%d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %-5p %c %L - %m%n"/>

2015-01-01 12:25:38.123 [main] DEBUG com.abc.ClassA 2 - debug msg
2015-01-01 12:25:38.123 [main] WARN com.abc.ClassA 3 - warn msg

7. 日誌紀錄完整度(1/3)

稽核

- 日誌紀錄包含以下項目
  - 1.識別使用者之ID(不可為個資類型)
  - 2.經系統校時後的時間戳記
  - 3.執行的功能或存取的資源
  - 4.事件類型或等級(priority)
  - 5.事件描述
- 建議驗證測試之範圍
  - 系統身分驗證機制
  - 系統主要服務功能
  - 管理者功能
  - 系統機敏資料新增或異動
  - 發生例外錯誤時

7. 日誌紀錄完整度(2/3)

稽核

- 使用者ID
  - 唯一值，避免共用帳號
  - 不可為個資類型，例如Email
- 時間
  - 系統設定自動校時
  - 記錄單位準確至微秒
- 執行的功能或存取的資源
  - 記錄方法(method)名稱或資源名稱
- 事件類型或等級(priority)
  - 區分事件類型，例如身分驗證、主要功能等
- 事件描述
  - 以文字描述該筆日誌紀錄之事件內容，避免重複使用

7. 日誌紀錄完整度(3/3)

稽核

- 事件發生當下物件資訊
  - 記錄相關除錯用物件資訊
  - 應避免記錄個資
- 網路來源與目的位址
  - 如有需要，記錄source ip、destination ip
- 錯誤代碼
  - 對錯誤進行分類並編製錯誤代碼表
  - 發生錯誤時顯示錯誤代碼，避免顯示過多資訊

8. 會談閒置自動失效機制(1/2)

會談管理

- 需求項目
  - 使用者的會談階段，設定該帳號在合理的時間(至多30分鐘)內未活動即自動失效

- 說明
  - 使用者於系統中若一段時間未進行活動，系統應有自動機制將該使用者的會談階段設為失效，以避免資安風險

8. 會談閒置自動失效機制(2/2)

會談管理

- 實作方法
  - 網站系統透過伺服器設定檔即可達成

```
<session-config>
  <session-timeout>30</session-timeout>
  <cookie-config>
    <secure>true</secure>
  </cookie-config>
</session-config>
```

- 驗證方法
  - 使用系統帳號，實際測試閒置超過設定時間後，再行動作時是否被登出回到首頁，且已無法執行原先動作

9. 會談手動失效機制(1/3)

會談管理

- 需求項目
  - 使用者的會談階段在登出後失效
- 說明
  - 系統應有手動機制，使用者明確進行登出後，將該使用者的會談階
    - 段設為失效

9. 會談手動失效機制(2/3)

- 實作方法
  - 以程式語言規定的方法，明確地將會談階段(Session)無效化

```
public class LogoutAction extends BaseAction {

  public String execute() throws IOException {
    setLoggingData(Logging.LOG_IN_OUT, "登出成功！");
    HttpSession session = ServletActionContext.getRequest().getSession();
    String logoutpage = (String) session.getAttribute("locale");
    session.invalidate();
    if ("en_US".equals(logoutpage)) {
```

9. 會談手動失效機制(3/3)

會談管理

- 驗證方法
  - 實際以帳號測試執行登出動作後，不再具有任何系統權限，無法存取非公開之網頁或功能

http://www.abc.com/profile.jsp

回到首頁

再次嘗試存取

10. 簡要錯誤訊息(1/3)

錯誤及例外處理

- 需求項目
  - 發生錯誤時，使用者頁面僅顯示簡短錯誤訊息及代碼，不包含詳細的錯誤訊息
- 說明
  - 系統應設計錯誤處理機制，當系統發生錯誤時，採取錯誤代碼或簡短訊息呈現，避免將詳細或除錯用訊息直接顯示於使用者頁面，以防被攻擊者用來刺探系統內部資訊，或根據錯誤訊息推測出系統可能之弱點

10. 簡要錯誤訊息(2/3)

- 實作方法
  - 1. 使用程式語言try-catch語法，在發生錯誤時導向錯誤訊息網頁並顯示特定訊息。
  - 2. 使用網站伺服器錯誤處理機制

```
<error-page>
  <exception-type>java.lang.Throwable</exception-type>
  <location>/error.jsp</location>
</error-page>
```

10. 簡要錯誤訊息(3/3)

- 驗證方法
  - 刻意於程式碼中產生錯誤，確認執行該功能時會導向錯誤訊息頁面

```
//使用者是否被鎖定
if( checkLockout(list, id, ip, period) ){
  request.setAttribute("msg", "使用者帳戶已被鎖定.");
  requestDispatcher.forward(request, response);
  return;
}

if(true)
  throw new Exception("Error occurred");

//檢查帳號密碼是否正確
if( checkPW(list, password) ){
  if (checkExpiration(list, "1")){
    HttpSession s = request.getSession();
    s.setAttribute("id", id);
```

11. 限制管理者介面(1/2)

組態管理

- 需求項目
  - 管理者介面限制存取來源或不允許遠端存取

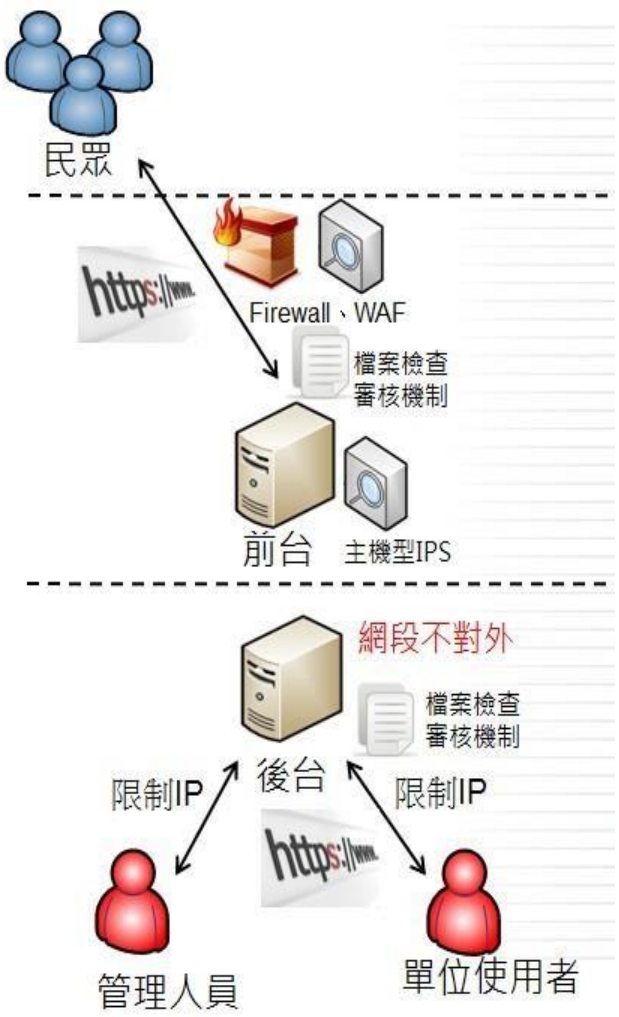
- 說明
  - 管理者介面通常可執行系統中較高權限的功能(例如權限與人員管理)，其相對風險較高，因此應盡可能不允許遠端存取，僅允許透過內部網路存取，以避免有心人士從外部嘗試攻擊之可能。若有必要允許外部遠端存取管理者介面，應限制特定存取來源IP，避免全面性開放存取

11. 限制管理者介面(2/2)

組態管理

- 實作方法
  - 區分前後台，將管理功能移至後台。
  - 後台限制僅允許內部來源
  - 管理者介面如須對外部網路開放，則必須限制僅允許特定IP存取

- 驗證方法
  - 以測試案例，實際驗證外部IP無法連線管理者介面



檔案下載網址

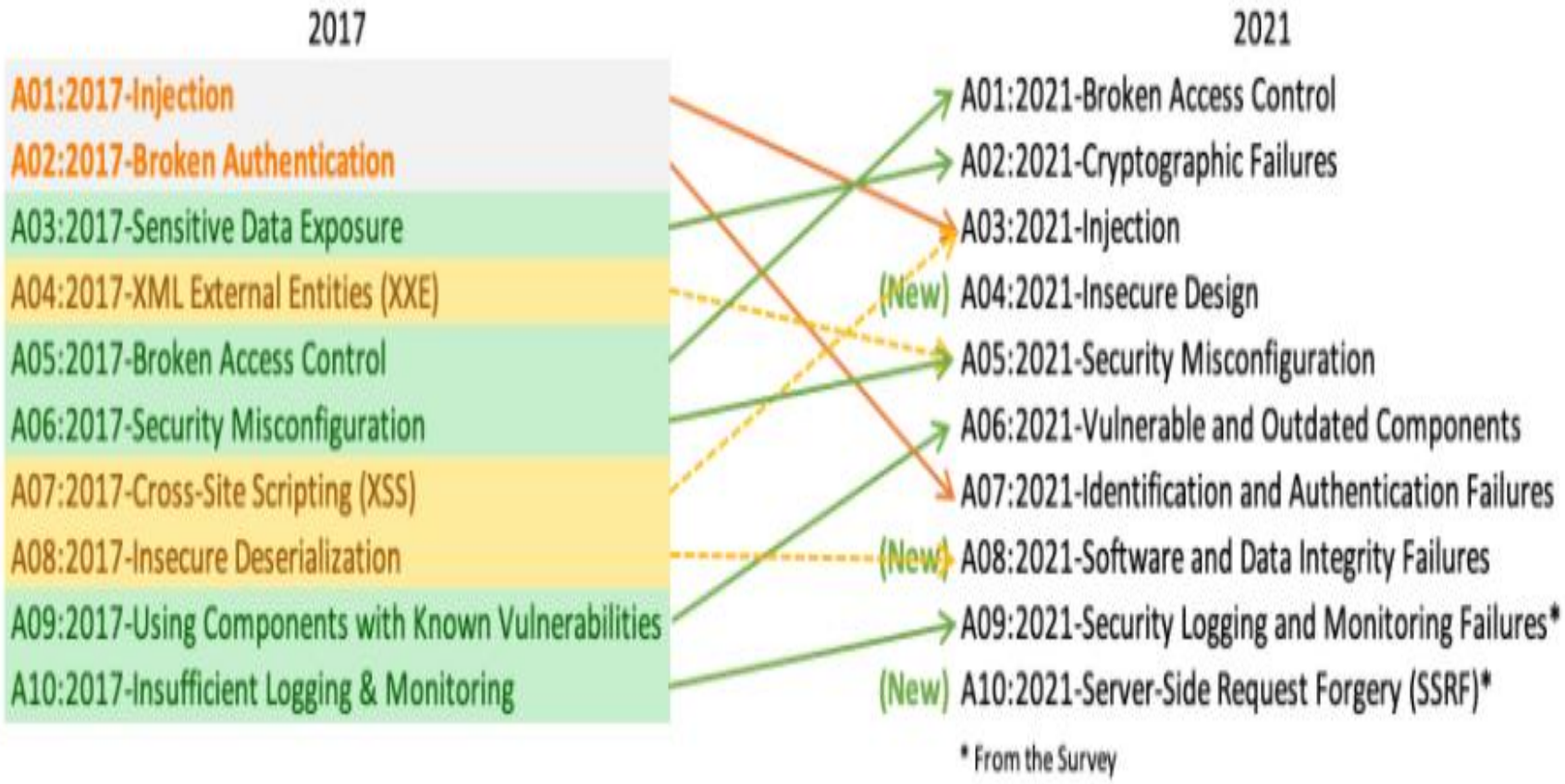
https://www.nccst.nat.gov.tw/HandoutDetail?lang=zh&seq=1253

OWASP簡介

● OWASP(The Open Web Application Security Project)是一個全球性、非
營利的組織，專注在增進軟體的安全

● 其下有眾多安全相關項目
– OWASP Top 10 (十大網站安全風險)
– OWASP Mobile Top 10 (十大行動裝置安全風險)
– Application Security Verification Standard (系統安全驗證)
– Software Assurance Maturity Model (軟體安全成熟度)
● OWASP Enterprise Security API (ESAPI)
– 是免費、開放原始碼的網頁安全控制函式庫，目的使程式設計師較容易地
寫出低風險的應用程式
– 具有Java、.NET、PHP、C、C++、Objective C、Javascript、Python、Perl等多種版本

OWASP 十大網站漏洞



安全的軟體實作

安全的軟體實作階段乃針對「輸入資料驗證」、「輸出資料確認」、「資料保護」、「驗證檢查」、「錯誤訊息處理」、「安全架構檔」及「源碼完整性」等7 類專案，透過範例說明如何實作。

輸入資料驗證

在用戶端實作輸入驗證可以協助使用者正確的填寫資料，同時可以降低技術人員的負擔，例如不會發生電話號碼欄位元出現英文字母、性別欄位元出現數字及資料庫出現攻擊字串等。雖然在用戶端實作輸入驗證可以提高使用者的便利性與降低系統被攻擊的風險，但是不適合用來做系統主要的安全驗證機制，因為實作在用戶端很容易被繞過，例如關掉瀏覽器執行 JavaScript 的功能與使用代理伺服器軟體，輸入驗證最好實作在伺服端，範例詳見圖 所示。

```
...
public String escapeShellCmd(String input) {
  String SPECIALCHAR = "#&;`|*?~<>^()[]{}$" + '\\' + '\'' + '"' + '\n';
                              String filteredInput = input;
  char strValidate;
  for (int j=0; j<SPECIALCHAR.length(); j++)
  {
     strValidate = SPECIALCHAR.charAt(j);
     filteredInput = filteredInput.replace(strValidate,' ');
  }
                              return filteredInput;
}
...
String dir = request.getParameter("dir");
Runtime runtime = Runtime.getRuntime();
Process proc = runtime.exec("/bin/sh du -s /webroot/" + escapeShellCmd(dir));
...
```

輸出資料確認

將沒經過編碼或跳脫的使用者輸入直接輸出到頁面，容易造成常見的跨網站入侵字串弱點，弱點描述與修補建議如下所示：

– 跨網站入侵字串(Cross Site Scripting)

跨網站入侵字串弱點(XSS)是一個允許攻擊者植入惡意程式碼並讓瀏覽器執行的弱點。它主要依賴Web 應用程式接受不可信任之字串，並將此作為輸出HTML 檔的內容之一，使得受害者的瀏覽器將被迫執行任意的腳本語言。此時，瀏覽器可能會允許這些惡意程式存取 cookies、session tokens 及其他該網站專屬的敏感資訊

輸出資料確認

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {

        public void doGet(HttpServletRequest request,
                          HttpServletResponse response)
      throws ServletException, IOException {

            PrintWriter out = response.getWriter();

            String name = request.getParameter("name");

            out.println(name);

   }
}
```

Cross Site Scripting JAVA錯誤範例

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.apache.commons.lang.StringEscapeUtils;


public class HelloWorld extends HttpServlet {

        public void doGet(HttpServletRequest request,
                          HttpServletResponse response)
      throws ServletException, IOException {

            PrintWriter out = response.getWriter();

            String name = request.getParameter("name");

            out.println(StringEscapeUtils.escapeHtml(name));

   }
}
```

Cross Site Scripting JAVA 修補建議範例

資料保護

機敏資料不能以明文的方式儲存，必須經過加解密機制處理，避免機敏資料外流，實務上常見的不安全儲存(Insecure Storage)弱點描述與修補建議如下所示：

– 不安全的儲存

明文密碼缺失(CWE 259)是將密碼直接儲存在程式碼中。這種撰寫風格會造成密碼管理上的困難，因為：

Ø所有專案的開發人員都可以看到源碼，也有機會知悉資料庫帳號密碼

Ø當程式碼外流時，資料庫帳號密碼也隨之外泄

Ø純文字密碼有可能被從已編譯過的程式碼中取出

Ø當軟體已進入生產，密碼無法被更改，除非是發行修正程式

Ø當密碼要更改時，每個包含密碼的檔案都必須跟著更新

資料保護

.資料保護驗證需求
資料保護驗證方法實作如下：
– 關閉用戶端的快取與自動完成功能
– 明定政策來規範如何控制機敏資料的存取
– 不可經由URL參數傳遞敏感性資料，必須存放在訊息本文內
– 傳送到用戶端的敏感性資料的快取或暫存必須受到保護
– 存放在伺服器內敏感性資料的快取或暫存必須受到保護

資料保護

trunk/KnowledgeSystem/src/cn/siox/util/database/DictOperator.java

```
73:   String url = "jdbc:mysql://localhost:3306/siox_english";
74:   java.sql.Connection conn = DriverManager.getConnection(url, "root", "111111");
75:   java.sql.Statement stmt = conn.createStatement();
```

siox-knowledge-kit.googlecode.com/svn - Unknown - Java - More from svn »

trunk/src/test/dbtest.java

```
130:  //
131:  con = DriverManager.getConnection(url, "root", "iiist");
132:  Statement select = con.createStatement( );
```

stockmanagementcapstoneproject.googlecode.com/svn - Unknown - Java

uEngine-Web/genericWebApp/selection.jsp

```
12:         String url="jdbc:oracle:thin:@172.16.2.163:1521:demoora";
13:         Connection conn = DriverManager.getConnection(url, "renewal", "renewal");
14:
```

uengine.cvs.sourceforge.net/cvsroot/uengine - Unknown - JSP - More from uengine »

trunk/QYProject/src/test/JDBCTest.java

```
48:   // 建立連接
49:   conn = DriverManager.getConnection(url, "hslj6", "hslj6");
50:   Statement stmt = conn.createStatement();
```

personaldatacollector.googlecode.com/svn - Unknown - Java - More from svn »

trunk/momobot/src/momobot/Db.java

```
104:  public static Connection getConnection() throws SQLException {
105:      return DriverManager.getConnection(URL, "momobot", "tobomom");
```

開放源碼套件的純文字密碼案例

資料保護

以 Java 為例，應該寫為透過存取Property來匯入資料庫帳號密碼。
例如：

```
Properties props = new Properties();
props.load(this.getClass().getResourceAsStream("/config/jdbc.properties
"));
String DB_DRIVER = props.getProperty("jdbc.driver");
String DB_URL = props.getProperty("jdbc.url");
String DB_USER = props.getProperty("jdbc.user");
String DB_PASSWORD = props.getProperty("jdbc.password");
Class.forName(DB_DRIVER);
Connection connection = DriverManager.getConnection(DB_URL,
DB_USER,
DB_PASSWORD);
```

純文字密碼JAVA 修補建議範例

驗證檢查

常見的身分認證弱點以SQL注入弱點最為知名，其發生在資料庫伺服器可以被用來執行外部的SQL命令時。一般來說都是由Web 應用程式的前端執行，這種攻擊包含輸入惡意組成的SQL語法，造成資料庫伺服器執行未授權的SQL命令，或繞過登入驗證程式以取得存取系統的許可權。

```
rs = stmt.executeQuery("select * from user where username='" + username + "' and password='" + password + "'");
PreparedStatement prepStmt = con.prepareStatement("SELECT * FROM user WHERE userId = '" + userName + "'");
ResultSet rs = prepStmt.executeQuery();
```

SQL Injection JAVA錯誤範例

```
String selectStatement = "SELECT * FROM User WHERE userId = ? ";
PreparedStatement prepStmt = con.prepareStatement(selectStatement);
prepStmt.setString(1, userId);
ResultSet rs = prepStmt.executeQuery();
```

SQL Injection JAVA 修補建議範例

會談管理

使用者通過驗證登入系統後，系統會透過會談管理來處理使用者的存取權限，由於系統並不瞭解實際存取之使用者是誰，只透過會談管理來與用戶端溝通，故必須確保會談管理之安全性與完整性，避免驗證後之許可權被濫用。會談管理必須考慮下列專案：

- 系統閒置一段時間後自動註銷
- 登出後必須清除所有已驗證內容

會談管理

```
...
session_start();

if(!session_is_registered("session_count")) {
        $session_count = 0;
        $session_start = time();
        $_SESSION['session_count']=$session_count;
        $_SESSION['session_start']=$session_start;
} else {
        $session_count++;
}

$session_timeout = 1800;

$session_duration = time() - $session_start;
if ($session_duration > $session_timeout) {
    session_unset();
    session_destroy();
    $_SESSION = array();
        header("Location: /login_page.php?expired=yes");
} else {
    $session_start = time();
    $_SESSION['session_start']=$session_start;
}
...
```

系統閒置後自動登出範例

存取控制驗證需求

系統必須限制使用者之存取權限，避免通過驗證之使用者任意使用系統命
令及資源而造成命令注入(Command Injection)、資源注入(Resource Injection)及檔案注入(File Injection)等問題。

– 命令注入

程式接收使用者輸入資料，並利用這些資料執行系統命令。在許多情況下，攻擊者可控制這些資料，藉以執行並非程式原先意圖執行命令。分隔字元(例如分號)被執行，這樣可允許攻擊者執行該程式可執行的命令。這種弱點可能導致系統與資料的安全性大幅降低，例如原先預定被執行的命令是由資料庫管理元件執行，攻擊者即可利用這個弱點刪除整個資料庫。防止這種攻擊最有效的方法，是確保使用者輸入字元經過過濾，僅接受特定的字元，如同其他弱點，白名單的方式要比黑名單來的好。

存取控制驗證需求

```
Runtime.getRuntime().exec(request.getParameter("c"));
```

命令注入JAVA 錯誤範例

```
public String filterInput(String input) {
        String filteredInput = input;
        filteredInput = filteredInput.replace ('!','_');
        filteredInput = filteredInput.replace ('$','_');
        filteredInput = filteredInput.replace ('^','_');
        filteredInput = filteredInput.replace ('&','_');
        filteredInput = filteredInput.replace ('*','_');
        filteredInput = filteredInput.replace ('(','_');
        filteredInput = filteredInput.replace (')','_');
        filteredInput = filteredInput.replace ('~','_');
        filteredInput = filteredInput.replace ('[','_');
        filteredInput = filteredInput.replace (']','_');
        filteredInput = filteredInput.replace ('\\','_');
        filteredInput = filteredInput.replace ('|','_');
        filteredInput = filteredInput.replace ('{','_');
        filteredInput = filteredInput.replace ('}','_');
        filteredInput = filteredInput.replace ('`','_');
        filteredInput = filteredInput.replace ('"','_');
        filteredInput = filteredInput.replace (';','_');
        filteredInput = filteredInput.replace ('<','_');
        filteredInput = filteredInput.replace ('>','_');
        filteredInput = filteredInput.replace ('?','_');
        filteredInput = filteredInput.replace ('-','_');
        return filteredInput;
          }
Runtime.getRuntime().exec(filterInput(request.getParameter("c")));
```

命令注入JAVA 修補建議範例

存取控制驗證需求

–資源注入

程式無正確檢查讀取檔案的來源時，攻擊者有可能上傳一個和開發人員預定完全不同的檔案，而讓攻擊者上傳惡意內容到Web 應用程式。一個常見會造成資源注入弱點的情況，是當遠端網站伺服器讀取一個檔案，並使用裡面的資料來進行資料庫查詢，或者以符合網站風格的方式顯示在網頁上。

存取控制驗證需求

```
File filePath=new File(request.getParameter(input));
```

資源注入JAVA 錯誤範例

```
public String filterDir(String input) {
        String filteredInput = input;

        filteredInput = filteredInput.replace ('.','_');
        filteredInput = filteredInput.replace ("..","_");
        filteredInput = filteredInput.replace ('"','_');
        filteredInput = filteredInput.replace ('\\','_');
        filteredInput = filteredInput.replace ('%','_');
        filteredInput = filteredInput.replace (';','_');
        filteredInput = filteredInput.replace ('(','_');
        filteredInput = filteredInput.replace (')','_');
        filteredInput = filteredInput.replace ('&','_');
        filteredInput = filteredInput.replace ('/','_');

        return filteredInput;
      }
File filePath=new File(filterDir(request.getParameter(input)));
```

資源注入JAVA 修補建議範例

存取控制驗證需求

–檔案注入
惡意的使用者在有弱點的網站上執行自己的程式碼。這個攻擊是依靠在沒有檢查輸入的程式中，引用惡意的程式碼。如果攻擊者能將他們的惡意程式碼注入到一個網頁中，就有可能利用PHP 腳本去引用一個遠端檔案，而不是原先信任的本機檔案。一個常見的檔案注入弱點情況，是當開發人員讓程式開啟一個本機檔案，可能是作為頁首或頁尾的一個範本

存取控制驗證需求

URL也可以被使用在include()，include_once()，require()和require_once()敘述中。

```
require('article2'.$_REQUEST["n"].'.php');
```

檔案注入PHP 錯誤範例

```
require('article2'.basename($_REQUEST["n"]).'.php');
```

檔案注入PHP 修補建議範例

通訊安全驗證需求

避免將不可信任的使用者輸入資料視為網頁轉址的目的地，降低使用者在系統頁面轉換過程中被導到惡意頁面的風險。

```
...
String strDest = request.getParameter("dest");
response.sendRedirect(strDest);
...
```

任意轉址 JAVA 錯誤範例

```
...
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

        String strDest = request.getParameter("dest");
        response.sendRedirect(checkRedirectDest(strDest));
    }
    public String checkRedirectDest(String destStr)
        throws UnsupportedEncodingException {
        destStr = URLDecoder.decode(destStr, "UTF-8");
        if (destStr.startsWith("/") || destStr.startsWith("http://www/")) {
            return destStr;
        } else {
            return "/" + destStr;
        }
    }
}
...
```

任意轉址 JAVA 修補建議範例

HTTP 安全驗證需求

未經驗證的資料被寫入HTTP 標頭時，這樣可能會允許攻擊者設定整個傳送到瀏覽器的HTTP 應答。目標Web 應用程式允許輸入包含CR (carriagereturn，也被表示成%0d 或\r)與LF (line feed, 也被表示成%0a 或\n)字元被加入到標頭中。這些字元不僅讓攻擊者控制Web 應用程式原本要傳送的標頭與內容，也允許攻擊者任意加入其他的應答。

```
...
String tainted = request.getParameter("tainted");
Cookie cookie = new Cookie("123",tainted);
response.addCookie(cookie);
...
```

Http 應答分割JAVA錯誤範例

```
...
String safe = URLEncoder.encode(request.getParameter("tainted"));
Cookie cookie = new Cookie("123",safe);
response.addCookie(cookie);
...
```

Http 應答分割JAVA 修補建議範例

安全設定驗證需求

設定檔在實務上較容易成為三不管地帶，MIS 人員以為開發人員最清楚設定檔應如何配置，開發人員以為MIS 人員最瞭解部署環境應如何設定，結果雙方因缺乏溝通而都以為設定檔已妥善處理，但實務上有諸多議題需要雙方互相討論、交流並形成共識。例如：

– 設定檔應該要儲存在何處
– 設定檔應該配置何種存取權限
– 設定檔裡面是否有機敏資訊
– 這些機敏資訊是否需要加密

連接到資料庫的相關設定不能以明文的方式存放於程式碼，建議透過設定檔來連接資料庫與避免將設定檔存放在Web 應用程式的根目錄下，防止攻擊者利用目錄遊走(path traversal)的弱點取得設定檔。建議使用加密保護並透過許可權控管機制限制不當的存取設定檔。

內部安全驗證需求

存取Web 應用程式之機敏資源都必須經過驗證，例如：沒有經過驗證或權限不足的使用者不得存取需授權才能存取之設定檔、程式碼及檔等，以避免機敏資料外流或不當存取。實作驗證檢查規範的所有專案以降低非法存取風險。

錯誤訊息處理

過於詳細的錯誤訊息或除錯資訊，例如Web Server 版本、資料庫版本及發生錯誤的行數等，會提高系統被入侵的風險，因此必須考慮異常錯誤處理機制，包括：

– 編譯時，是否包含debug 信息
– 程式碼中是否有處理錯誤與異常情況
– 程式碼如果沒有處理，那是否有全域與預設的處理方式，例如：.NET中的global exception handler
– 當錯誤發生時，是否會洩漏機敏資訊

```
ex.printStackTrace();
```

不當錯誤資訊揭露JAVA錯誤範例

```
ExitError("Error: $file does not exist");
```

不當錯誤資訊揭露JAVA 修補建議範例

人為錯誤之類型

- 人為錯誤十大類：
  1.健忘
  2.因誤解所造成的錯誤
  3.識別錯誤
  4.專業不足
  5.無心之過
  6.疏忽的錯誤
  7.因遲緩造成的錯誤
  8.缺乏標準/管理造成的錯誤
  9.意外的錯誤
  10.故意的錯誤

- 十項保障措施:
  1.預先警告操作員工、定期檢查
  2.訓練與標準化
  3.培訓、留意、警惕
  4.技能建立、標準化
  5.基礎教育、經驗
  6.留意、紀律、標準化
  7.技能建立、標準化
  8.工作指導、標準化
  9.全面生產性維護
  10.紀律、基礎知識教育

網站安全性管理策略與技術

安全程式的開發邏輯:

- 套件使用安全性注意原始碼內容是否有異常語法。
- 引用網路元件時盡量改變原始內容檔案目錄結構、程式結構、資料庫連結路徑等。
- 提供開發者所使用訊息不要成為公開資訊。
- 檔案、資料夾名稱使用非一般常用命名。
- 程式開發可使用元件方式代替字串變數，將常用變數可以寫成元件，程式可用呼叫元件方式，增加入侵的困難度。
- 程式開發前可先期規劃架構，可以網站劃分成畫面部份、畫面連結(action)部份、資料庫連結(action)部份及資料庫的部份。

資通安全防護及控制措施

- 資訊及資通系統之管理
  - 資訊及資通系統之使用
    - 使用資訊及資通系統前應經其管理人授權
    - 使用資訊及資通系統時，應留意其資通安全要求事項，並負對應之責任
    - 對於資訊及資通系統，宜識別並以文件記錄及實作可被接受使用之規則
  - 資訊及資通系統之刪除或汰除
    - 刪除或汰除前應評估機關是否已無需使用該等資訊及資通系統
    - 刪除或汰除時宜加以清查，以確保所有機敏性資訊及具使用授權軟體已被移除或安全覆寫
    - 具機敏性之資訊或具授權軟體之資通系統，宜採取實體銷毀，或以毀損、刪除或覆寫之技術，使原始資訊無法被讀取，並避免僅使用標準刪除或格式化功能

資通安全防護及控制措施

- 存取控制與加密機制管理

網路安全控管

- 定期檢視防火牆政策是否適當，並適時進行防火牆軟、硬體之必要更新或升級
- 網路區域應進行區隔(外部網路、非軍事區、內部網路)，各區域間連線需經防火牆進行存取控制，非允許的服務與來源不能進入其他區域
- 對於通過防火牆之來源端主機IP位址、目的端主機IP位址、來源通訊埠編號、目的地通訊埠編號、通訊協定、登入登出時間、存取時間以及採取的行動，均應予確實記錄
- DNS伺服器應設定指向GSN Cache DNS
- 機密資料原則不得透過無線網路及設備存取、處理或傳送

資通系統權限管理

- 資通系統應設置通行碼管理，通行碼之要求需滿足
  (1)通行碼長度8碼以上
  (2)通行碼複雜度應包含英文大小寫、特殊符號或數字兩種以上
  (3)使用者每90天應更換通行碼
- 使用資通系統前需經授權，並使用唯一之使用者ID，除有特殊營運或作業必要經核准並紀錄外，不得共用ID
- 無繼續使用資通系統時，應立即停用或移除使用者ID，資通系統管理者應定期清查使用者之權限

資通安全防護及控制措施

- 存取控制與加密機制管理

特權帳號之存取管理

- 資通系統之特權帳號請應經正式申請授權方能使用，特權帳號授權前應妥善審查其必要性，其授權及審查記錄應留存
- 資通系統之特權帳號不得共用
- 對於特權帳號，宜指派與該使用者日常公務使用之不同使用者ID
- 資通系統特權帳號應妥善管理，並應留存特殊權限帳號使用軌跡
- 資通系統之管理者每季應清查系統特權帳號並劃定特權帳號逾期之處理方式

加密管理

- 加密保護措施應遵守下列規定
  (1)應落實使用者更新加密裝置並備份金鑰
  (2)應避免留存解密資訊
  (3)一旦加密資訊具遭破解跡象，應立即更改
- 機密資訊於儲存或傳輸時應進行加密

資通安全防護及控制措施

- 作業與通訊安全管理

防範惡意軟體控制措施

- 主機及個人電腦應安裝防毒軟體，並時進行軟、硬體之必要更新或升級
- 使用者未經同意不得私自安裝應用軟體，管理者並應每半年定期針對管理之設備進行軟體清查
- 使用者不得私自使用已知或有嫌疑惡意之網站
- 設備管理者應定期進行作業系統及軟體更新，以避免惡意軟體利用系統或軟體漏洞進行攻擊

遠距工作之安全措施

- 針對遠距工作之連線應採適當之防護措施(並包含伺服器端之集中過濾機制檢查使用者之授權)，並且記錄其登入情形
  (1)提供適當通訊設備，並指定遠端存取之方式
  (2)進行遠距工作時之安全監視
  (3)提供虛擬桌面存取，以防止於私有設備上處理及儲存資訊
- 資通安全推動小組應定期審查已授權之遠距工作需求是否適當
- 資通系統之操作及維護以現場操作為原則，避免使用遠距工作，如有緊急需求時，應申請並經資通安全推動小組同意後始可開通

資通安全防護及控制措施

- 作業與通訊安全管理

電子郵件安全管理

- 人員到職後應經申請方可使用電子郵件帳號，並應於人員離職後刪除電子郵件帳號之使用
- 電子郵件伺服器應設置防毒及過濾機制，並適時進行軟硬體之必要更新
- 電子郵件管理及使用規定如下
  (1)系統管理人應定期清查電子郵件帳號
  (2)避免讀取來歷不明之郵件或含有巨集檔案之郵件，以防範社交工程攻擊
  (3)確保電子郵件傳送時之傳遞正確性
  (4)注意電子簽章之要求事項
  (5)純文字模式閱覽

確保實體與環境安全措施

- 資料中心及電腦機房之門禁管理
  (1)機關人員或來訪人員應申請及授權後，方可進入。管理者並應定期檢視授權人員之名單
  (2)人員及設備進出資料中心及電腦機房應留存紀錄
- 資料中心及電腦機房之環境控制
  (1)應安裝安全偵測及防護措施
  (2)各項安全設備應定期執行檢查、維修
- 辦公室區域實體與環境安全措施
  (1)考量採用辦公桌面淨空政策
  (2)機密性及敏感性資訊，不使用或下班時應該上鎖

資通安全防護及控制措施

- 作業與通訊安全管理

資料備份

- 重要資料及核心資通系統應進行資料備份，其備份之頻率應滿足復原時間點目標之要求，並執行異地存放
- 每季確認核心資通系統資料備份之有效性。且測試該等資料備份時，宜於專屬之測試系統上執行，而非直接於覆寫回原資通系統
- 備份資料如有機密性考量，宜加密保護

媒體防護措施

- 使用隨身碟或磁片等存放資料時，具機密性、敏感性之資料應與一般資料分開儲存，不得混用並妥善保管
- 資訊如以實體儲存媒體方式傳送，應留意實體儲存媒體之包裝，選擇適當人員進行傳送，並應保留傳送及簽收之記錄
- 對機密與敏感性資料之儲存媒體實施防護措施，包含機密與敏感之紙本或備份磁帶，應保存於上鎖之櫃子，且需由專人管理鑰匙

資通安全防護及控制措施

- 作業與通訊安全管理

電腦使用之安全管理

- 禁止私自安裝點對點檔案分享軟體及未經合法授權軟體
- 電腦應隨時配合更新作業系統、應用程式漏洞修補程式及防毒病毒碼等
- 下班時應關閉電腦及螢幕電源
- 電腦若超過十五分鐘不使用時，應立即登出或啟動螢幕保護功能

行動設備之安全管理

- 機密資料不得由未經許可之行動設備存取、處理或傳送
- 機敏會議或場所不得攜帶未經許可之行動設備進入

資通安全防護及控制措施

- 作業與通訊安全管理

即時通訊軟體安全管理

- 使用即時通訊軟體傳遞機關內部公務訊息，其內容不得涉及機密資料。但有業務需求者，應使用經專責機關鑑定相符機密等級保密機制或指定之軟、硬體，並依相關規定辦理
- 使用於傳遞公務訊息之即時通訊軟體應具備下列安全性需求
  (1)用戶端應有身分識別及認證機制
  (2)訊息於傳輸過程應有安全加密機制
  (3)應通過經濟部工業局訂定行動化應用軟體之中級檢測項目
  (4)伺服器端之主機設備及通訊紀錄應置於我國境內
  (5)伺服器通訊紀錄（log）應至少保存六個月


THE
END

A1 – Injection（注入攻擊）

網站應用程式執行來自外部包括資料庫在內的惡意指令，SQL Injection與Command Injection等攻擊包括在內。因為駭客必須猜測管理者所撰寫的方式，因此又稱「駭客的填空遊戲」。

舉例來說，原本管理者設計的登入頁面資料庫語法如下：

$str = "SELECT * FROM Users WHERE Username='".$user."' and Password='".$pass."'";

如果說$user以及$pass變數沒有做保護，駭客只要輸入「' or ''='」字串，就會變成以下：

*$str = "SELECT * FROM Users WHERE Username='' or ''='' and Password='' or ''='' ";*

如此一來，這個SQL語法就會規避驗證手續，直接顯示資料。

A1 – Injection（注入攻擊）

簡述駭客攻擊流程：
找出未保護變數，作為注入點
猜測完整Command並嘗試插入
推測欄位數、Table名稱、SQL版本等資訊
完整插入完成攻擊程式

防護建議：
使用Prepared Statements，例如Java PreparedStatement()，.NET SqlCommand(), OleDbCommand()，PHP PDO bindParam()
使用Stored Procedures
嚴密的檢查所有輸入值
使用過濾字串函數過濾非法的字元，例如mysql_real_escape_string、addslashes
控管錯誤訊息只有管理者可以閱讀
控管資料庫及網站使用者帳號許可權為何

A2 – Cross Site Scripting (XSS) (跨站腳本攻擊)

網站應用程式直接將來自使用者的執行請求送回瀏覽器執行，使得攻擊者可擷取用戶的Cookie或Session資料而能假冒直接登入為合法使用者。

此為目前受災最廣的攻擊。簡稱XSS攻擊。
受害者登入一個網站
從Server端取得Cookie
但是Server端上有著XSS攻擊，使受害者將Cookie回傳至Bad Server
攻擊者從自己架設的Bad Server上取得受害者Cookie
攻擊者取得控制使用受害者的身分

A2 – Cross Site Scripting (XSS) (跨站腳本攻擊)

防護建議：

檢查頁面輸入數值

輸出頁面做Encoding檢查

使用白名單機制過濾，而不單只是黑名單

PHP使用htmlentities過濾字串

.NET使用Microsoft Anti-XSS Library

OWASP Cross Site Scripting Prevention Cheat Sheet

各種XSS攻擊的Pattern參考

A3 – Broken Authentication and Session Management（身分驗證功能缺失）

網站應用程式中自行撰寫的身分驗證相關功能有缺陷。例如，登入時無加密、SESSION無控管、Cookie未保護、密碼強度過弱等等。

例如：

應用程式SESSION Timeout沒有設定。使用者在使用公用電腦登入後卻沒有登出，只是關閉視窗。攻擊者在經過一段時間之後使用同一台電腦，卻可以直接登入。

網站並沒有使用SSL / TLS加密，使用者在使用一般網路或者無線網路時，被攻擊者使用Sniffer竊聽取得User ID、密碼、SESSION ID等，進一步登入該帳號。

這些都是身分驗證功能缺失的例子。

A3 – Broken Authentication and Session Management（身分驗證功能缺失）

管理者必須做以下的檢查：

所有的密碼、Session ID、以及其他資訊都有透過加密傳輸嗎？
憑證都有經過加密或hash保護嗎？
驗證資訊能被猜測到或被其他弱點修改嗎？
Session ID是否在URL中暴露出來？
Session ID是否有Timeout機制？

防護建議：

使用完善的COOKIE / SESSION保護機制
不允許外部SESSION
登入及修改資訊頁面使用SSL加密
設定完善的Timeout機制
驗證密碼強度及密碼更換機制

A4 – Insecure Direct Object References（不安全的物件參考）

攻擊者利用網站應用程式本身的檔案讀取功能任意存取檔案或重要資料。進一步利用這個弱點分析網站原始碼、系統帳號密碼檔等資訊，進而控制整台主機。

例如：
http://example/read.php?file=../../../../../../../c:\boot.ini。

防護建議：
避免將私密物件直接暴露給使用者
驗證所有物件是否為正確物件
使用Index / Hash等方法，而非直接讀取檔案

A5 – Cross Site Request Forgery (CSRF)（跨站冒名請求）

已登入網站應用程式的合法使用者執行到惡意的HTTP指令，但網站卻當成合法需求處理，使得惡意指令被正常執行。

舉例來說，攻擊者在網站內放置了 *<img src=”http://server.com/send.asp?to=...”>* ，受害者讀取到此頁面之後，就會去server.com主機執行send.asp惡意行為。

例如Web 2.0時代的社交網站等等，都是CSRF攻擊的天堂。

防護建議：

確保網站內沒有任何可供XSS攻擊的弱點
在Input欄位加上亂數產生的驗證編碼
在能使用高許可權的頁面，重新驗證使用者
禁止使用GET參數傳遞防止快速散佈
使用Captcha等技術驗證是否為人為操作

A6 – Security Misconfiguration（安全性設定疏失）

系統的安全性取決於應用程式、伺服器、平臺的設定。因此所有設定值必須確保安全，避免預設帳號、密碼、路徑等。甚至被Google Hacking直接取得攻擊弱點。

防護建議：

軟體、作業系統是否都有更新到最新版本？是否都有上最新Patch?
不需要的帳號、頁面、服務、埠是否都有關閉？
預設密碼是否都有更改？
安全設定是否都完備？
伺服器是否都有經過防火牆等設備保護？

各種設備、系統的預設密碼，都可以在網路上找到一些整理資料。

http://www.phenoelit-us.org/dpl/dpl.html
http://www.routerpasswords.com/
http://www.defaultpassword.com/

A7 – Failure to Restrict URL Access（限制URL存取失敗）

網頁因為沒有許可權控制，使得攻擊者可透過網址直接存取能夠擁有許可權或進階資訊的頁面。例如管理介面、修改資料頁面、個人機敏資訊頁面洩漏等等。

舉例來說，

/admin
/backup
/logs
/phpmyadmin
/phpinfo.php
/manage

這些都是常見的路徑及檔案。攻擊者只要猜測到，就可以操弄主機。

防護建議：

HTTP Service直接限制來源IP
使用防火牆阻擋
密碼授權加密頁面
網站架構優化

A8 – Unvalidated Redirects and Forwards（未驗證的導向）

網頁應用程式經常將使用者Forward或Redirect至其他頁面或網站，沒有驗證的機制。攻擊者可將受害者導向至釣魚網站或惡意網站，甚至存取受限制的資源。

例如：

http://example.cc/redir.jsp?url=evil.com
http://example.cc/func.jsp?fwd=admin.jsp
http://g.msn.com/9SE/1?http://xxx.com

防護建議：

非必要時避免使用Redirect及Forward
驗證導向位置及存取資源是合法的

A9 – Insecure Cryptographic Storage（未加密的儲存設備）

網站應用程式沒有對敏感性資料使用加密、使用較弱的加密演算法或將金鑰儲存於容易被取得之處。加密演算法是安全防護的最後一道防線，當駭客取得了帳號密碼，可以簡單地使用一些破解軟體甚至線上服務進行破解。例如Cain & Abel，MD5 Reverse Lookup等。

防護建議：

使用現有公認安全的加密演算法

減少使用已有弱點的演算法，例如MD5 / SHA-1，甚至更簡單的加密法

安全的保存私密金鑰

A10 – Insufficient Transport Layer Protection（傳輸層保護不足）

網頁應用程式未在傳輸機敏資訊時提供加密功能，或者是使用過期、無效的憑證，使加密不可信賴。

例如：攻擊者竊聽無線網路，偷取使用者cookie；網站憑證無效，使用者誤入釣魚網站。

防護建議：

盡可能的使用加密連線

Cookie使用Secure Flag

確認加密憑證是有效並符合domain的

後端連線也使用加密通道傳輸

http://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

Cookie Secure Flag設定：

PHP setcookie ("TestCookie", "", time() - 3600, "/", ".example.com", 1);

JSP cookie.setSecure(true);

ASP.NET cookie.Secure = True;