附件3　WEB 網站建置與個人資料管理維運」RFP 資安需求範例

附錄 1　政府 Web 網站委外安全注意事項與安全檢核表

1. Web 應用程式的網路攻擊層出不窮

各機關目前委外開發之 Web 應用程式，在功能面與內容部分，大多能滿足民眾需求，惟在委外開發、購買及維護這類的應用程式方面，對於資安方面要求非常有限；近日有關 Web 應用程式的網路攻擊層出不窮，破壞技術與日俱增，影響日益嚴重；此外，個人資料保護法通過後，個人資料洩漏涉及的損害賠償責任很重,各機關透過網站公布的資訊常與個人資料相關，其資安控制措施如不加強，未來將面臨巨大挑戰。

2. 對 Web 應用程式安全漏洞的了解不足

鑑於網路技術發展日新月異，各機關資安工作的負擔日益沉重，在人力有限的情形下，對於 Web 應用程式安全漏洞的了解與相關的防範技術能力可能有所不足，因此特就 OWASP(Open Web Application Security Project - OWASP)所提供之開放原始碼計畫，各地分會製作出免費、公正、開放來源文獻之工具與標準，提供各機關參考。

3. 常見最嚴重的 Web 應用程式十大漏洞

OWASP 於 2017 年公布常見最嚴重的十大漏洞與風險，說明如下：

- 注入攻擊(Injections)

  透過資料庫語法漏洞與非完整之驗證機制，取得資料庫管理權限，並進行任意修改或是透過命令列之執行(如 Cmd.asp),最終取得主機管理權限,透過這個攻擊手法可直接下達系統命令，再經由此受害主機向內部其餘系統進行攻擊與資料竊取。

  注入式漏洞攻擊允許駭客利用轉送方，將惡意程式碼由一個網頁應用程式轉送到另一個系統中。這類型的攻擊包含利用系統呼叫程式的方式呼叫

作業系統，運用外部指令呼叫後端的伺服器，如 SQL Injection 就是運用 SQL 指令呼叫資料庫而形成攻擊。運用 Perl、Python 或其他語言撰寫的 Script 都可以輕易的注入設計不良的網站並執行這個 Script。任何運用直譯語言的網頁都會有注入式漏洞攻擊的風險。

許多網頁應用程式利用作業系統本身的功能或外部程式來提供一些服務，Send mail 就是一個最常被用到的外部程式。當一個網站傳送一個外部的 HTTP 請求時，就必須要小心，以免被駭客利用修改 HTTP 請求時加入惡意程式碼，做為注入攻擊的目標。

SQL Injection 就是這類攻擊最廣為流傳且最典型的。要利用 SQL Injection 的漏洞，駭客必須先找出後端有運用資料庫的網頁應用程式，然後小心地注入 SQL 惡意程式碼，使得這段程式碼可以通過網頁服務直達後端的資料庫，進行惡意的查詢。這類的攻擊並不難且有許多工具可以輔助找到這類的弱點，這類攻擊所造成的後果會使攻擊者有機會得到資料庫的存取權限、竄改資料庫或迫害資料庫的內容。

- 遭破壞的認證與連線管理(Broken Authentication and Session Management)

  認證與連線管理包括處理用戶鑑別與管理正在連線的會談(Session)。認證管理的流程是非常重要的部分，但就算是強大的認證機制也會被有問題的訊息認證管理所破壞，例如：改變密碼、忘記我的密碼、帳號更新及其他相關功能。對於很多 Web 應用程式來說，由於「過路人(Walk by)」攻擊可能會發生，所有的帳號管理功能都應該要求重新驗證，即使用戶持有有效的會談 ID。

  網頁的認證機制通常包含使用者帳號(ID)與密碼，而其他較強的認證機制可能會包含成本較高的軟硬體加密演算法與生物辨識法。一個廣泛使用的帳號再加上連線的漏洞，會造成使用者或管理者帳號被竊取或入侵。這是因為開發人員低估設計鑑別與連線管理間之複雜性，造成無法有效保護網站存取控制。

Web 應用程式必須建立會談連線以保存每個用戶的網頁要求。HTTP 本身不提供這樣的功能，所以 Web 應用程式必須自行建立。通常 Web 應用服務環境平台提供會談連線(Session)的能力，但是許多程式開發人員更喜歡建立自己的會談符記(Session Token)。不管哪一種情況，如果會談憑證沒有好好保護，攻擊者就可以攔截一個正在進行中的連線，從中竊取使用者的身分資料。除非所有的驗證訊息與連線鑑別皆被 SSL 保護，並且不暴露其他漏洞(如跨網站的入侵漏洞)，不然攻擊者仍可攔截一個正在連線的對話，冒充用戶的身分。因此，開發一個擁有強而有力連線憑證的架構，並保護這個連線憑證，將是程式開發者在整個網頁應用程式開發過程中所無法避免的。

- 跨網站腳本攻擊(Cross Site Scripting - XSS)

  跨網站腳本攻擊，或稱為 XSS，是以漏洞通常會發生在攻擊者利用一個網頁應用程式來傳送惡意程式碼，產生一個描述程式(Script)的字串並送給不同的使用者。這些漏洞會被廣為流傳且可發生在任何的網站應用程式上。

  攻擊者能夠使用跨網站的入侵字串傳送非法描述程式給沒有警覺的使用者。使用者的瀏覽器沒有辦法判斷是否該信任這個描述程式，而導致描述程式被執行。透過這個攻擊手法可取得使用者如帳號、密碼及信用卡等機密資訊，並同時植入間諜程式於用戶端上。

  XSS 的攻擊可以區分為兩類：儲存(Stored)與反射(Reflected)。儲存攻擊是指那些注入程式碼已經永久存在於目標伺服器中，如：資料庫伺服器、訊息論壇、訪客紀錄及命令欄位等。受害者會在請求這些伺服器中的資料時，無意中從伺服器取出惡意程式碼。反射攻擊是指那些注入程式碼並不存在於網頁伺服器中，如：錯誤訊息、搜尋結果或其他送到伺服器的請求後所得到的回應結果。反射攻擊會運用其他的路徑攻擊受害者，如：電子郵件或其他網頁伺服器。當使用者被設計，誤點含有惡意程式

碼的連結或送出一個含有惡意程式碼的表格，這些程式碼會傳送到有弱點的網頁伺服器，然後移轉到使用者的瀏覽器中並執行之。

- XML 外部實體(XML External Entities, XXE)

  由於 XML 處理器被設定允許引用 XML 檔案中的外部實體，使攻擊者可以上傳 XML 並在 XML 檔案中包含惡意內容，利用外部實體弱點竊取內部檔案或共用檔案、監聽內部掃描埠、執行遠端代碼和實施拒絕服務攻擊等。

- 斷開的訪問控制(Broken Access Control)

  當開發人員暴露一個對內部實現物件的引用時，例如，一個檔案(File)、目錄(Directory)或者資料庫密鑰(Database key)，就會產生一個不安全的直接物件引用(References)。在沒有檢查存取控制或其他保護時，攻擊者會利用這個引用去取得未授權資料。

  大多數 Web 應用程式應該在每個功能被訪問前在伺服器端執行相同的存取控制檢查。如果請求沒有被驗證，攻擊者能夠偽造請求以在未經適當授權時訪問功能。

- 不安全的設定管理(Insecure Configuration Management)

  網頁伺服器與應用程式的設定值在資訊安全中占有關鍵的地位。這些伺服器是用來提供內容或產生內容以服務使用者。此外，這些伺服器也可能提供資料儲存、訊息傳送、電子郵件及目錄服務等。一旦設定管理上有問題，將會廣泛的影響到整個網頁服務的運作。

  一般而言，網站程式開發者與網站管理者通常都隸屬於不同的部門，因此常會造成相當大的落差，而網頁應用程式的安全問題，通常跟這個落差有相當大的關聯，以下列出常見的因設定管理所造成的安全問題：

  - 未修補的網頁安全漏洞。
  - 網頁軟體的漏洞或錯誤的設定，造成檔案目錄外洩與目錄存取攻擊。

- 不必要的預設、備份、樣本檔案，包括 Scripts、應用程式、設定檔及網頁檔。
- 不適當的檔案與目錄權限。
- 開啟不必要的服務，包含網頁內容管理與遠端管理。
- 預設的使用者帳號與密碼。
- 開啟不必要的管理者功能或除錯功能。
- 過當的錯誤訊息顯示(錯誤訊息顯示過多的細節)。
- 使用自己發送的鑑別機制與中間者攻擊保護。
- 使用預設的鑑別機制。
- 不安全的外部鑑別系統。
- 錯誤的 SSL 鑑別與加密設定。

• 敏感資訊洩漏(Sensitive Data Exposure)

許多 Web 應用程式沒有正確保護敏感性資料，如信用卡，稅務 ID 和身分驗證憑證。攻擊者可能會竊取或篡改這些弱保護的資料以進行信用卡詐騙、身分竊取，或其他犯罪。

敏感性資料需額外的保護，比如在存放或在傳輸過程中的加密，以及在與瀏覽器交換時進行特殊的預防措施。

• 不安全的反序列化(Insecure Deserialization)

不安全的反序列化會導致遠端代碼執行，攻擊者可利用惡意或者竄改過的物件執行重播(replay)、注入和特權升級等攻擊。

• 利用已知弱點的元件(Using Known Vulnerable Components)

網站元件，例如函式庫(libraries)、框架(framework)和其它軟體模組(software modules)，幾乎總是以最大權限執行。如果一個帶有漏洞的元件被利用，這種攻擊可以造成嚴重的資料丟失或伺服器接管。應用程式使用帶有已知漏洞的元件會破壞應用程式防禦系統，並使一系列可能的攻擊和影響成為可能。

- 記錄和監測不足(Insufficient Logging & Monitoring)

  指 Web 應用程式進行日誌記錄、監控以及事件回應之缺失或無效，使攻擊者能夠進一步攻擊系統、保持持續性或轉向更多系統，以及竄改、提取或銷毀資料。

4. 安全對策

以上這些設定或設計上的錯誤都是我們常見會發生的，其中有許多必須由程式開發者與網站管理者共同研擬對策才能避免，因此兩者從程式開發階段就必須緊密的合作。

5. 風險管理

有些風險並非程式漏洞所造成的，例如「阻絕服務攻擊」是屬於一種網路攻擊行為，與應用程式安全與否無關，但對於提供網路服務確實是一項系統的風險。而「政府資訊作業委外安全參考指引」內容系依照 CNS27001 標準在整體安全風險的管理上著墨，故「政府資訊作業委外安全檢核表」著重於管理面的檢核，適用於各類委外服務。

但是所有在網路上提供便民服務的資訊系統，均會面對來自網際網路的各種風險，屬於系統發展類的軟體開發或系統整合案，無論系統大小或機密等級高低，招標過程大同小異，委外作業所要管理的風險差異不大，故難以依照系統之機密(防護)等級區分。

6. web **應用程式安全檢核**

為便於機關掌握 Web 應用程式安全檢核重點，篩選出具實務意義的重要控制措施項目共 75 項，依據系統安全等級區分「普」、「中」及「高」3 個等級，列舉適用的控制措施項目，方便各機關在辦理 web 應用程式委外開發時，依據系統的安全強度，要求廠商(或第三方)執行不同程度的檢核，控制措施詳細內容請參閱「Web 應用系統安全參考指引」，該指引亦針對各控制

措施以市面上常見之程式語言 ASP.NET、PHP 及 Java 提供範例。檢核內容詳見表 1「Web 應用程式安全檢核表」。

表1　Web 應用程式安全檢核表

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	◎	◎	◎	
稽核與可歸責性	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	◎	◎	◎	
		應稽核資訊系統管理者帳號所執行之各項功能	◎	◎	◎	
	稽核紀錄內容	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				
		採用單一的日誌紀錄機制，確保輸出格式的一致性	◎	◎	◎	
	稽核儲存容量	依據稽核紀錄儲存需求，配置稽核紀錄所需之儲存容量	◎	◎	◎	
	稽核處理失效之回應	資訊系統應在稽核處理失效(如儲存容量不足)之情況下，採取適當之行動，例如：關閉資訊系統、覆寫最舊的稽核紀錄或停止產生稽核紀錄等。	◎	◎	◎	
		當機關規定需要即時通報的稽核失效事件發生時，資訊系統應在機關規定之時效內，對機關特定之人員、角色提出告警(適用於高等級)			◎	
	時戳	資訊系統應使用系統內部時鐘產生稽核紀錄所需時戳，並可以對映到世界協調時間(UTC)或格林威治標準時間(GMT)	◎	◎	◎	
		系統內部時鐘應具備定期同步機制	◎	◎	◎	
	稽核資訊之保護	對日誌紀錄進行適當保護及備份，避免未經授權存取	◎	◎	◎	
		定期備份稽核紀錄到與原稽核系統不同之實體系統 (如 Log 伺服器)	◎	◎	◎	
		重要系統資料或紀錄留存雜湊值以確保完整性			◎	
營運持續計畫	資訊系統備份	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	◎	◎	◎	
	資訊系統備援	採用「高可用性」(High Availability) 架構(分散式或叢集伺服器架構)			◎	

識別與鑑別	內部使用者之識別與鑑別	採用多重因素身分鑑別(兩種以上驗證類型)			◎	
		資訊系統在建立連線前，應識別允許存取之特定來源(如 IP)			◎	
	身分鑑別管理	確實規範使用者密碼強度(密碼長度 12 個字元以上、包含英文大小寫、數字，以及特殊字元)	◎	◎	◎	
		使用者必須定期更換密碼，且至少不可以與前 3 次使用過之密碼相同	◎	◎	◎	
		具備帳號鎖定機制，帳號登入進行身分鑑別失敗達 5 次後，至少 15 分鐘內不允許該帳號及來源 IP 繼續嘗試登入	◎	◎	◎	
		身分鑑別相關資訊不以明文傳輸	◎	◎	◎	
		採用圖形驗證碼(CAPTCHA)機制於身分鑑別及重要交易行為，以防範自動化程式之嘗試		◎	◎	
		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)，檢查傳回令牌有效性後，才允許使用者進行重設密碼動作		◎	◎	
	鑑別資訊回饋	資訊系統應遮蔽在鑑別過程中之資訊(如密碼)，以防止未授權之使用者可能之窺探/使用	◎	◎	◎	
	加密模組鑑別	密碼添加亂數(Salt)進行雜湊函式(HASH Function)處理後，分別儲存亂數及雜湊後密碼		◎	◎	
系統與服	安全系統發展生命週期需求階段	針對系統安全需求，以檢核表方式進行確認	◎	◎	◎	
	安全系統發	應根據系統功能與要求，識別可能影響		◎	◎	

務獲得	展生命週期設計階段	系統之威脅，進行風險分析與評估				
		將風險評估結果回饋需求階段的檢核項目，並提出安全需求修正		◎	◎	
	安全系統發展生命週期開發階段	具有防範 SQL 命令注入攻擊(SQL Injection)之措施	◎	◎	◎	
		具有防範跨站腳本攻擊(XSS, Cross-Site Scripting)之措施	◎	◎	◎	
		具有防範「跨站請求偽造」(Cross-Site Request Forgery, CSRF)攻擊之措施	◎	◎	◎	
		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	◎	◎	◎	
		所有功能皆進行錯誤及例外處理，並確保將資源正確釋放	◎	◎	◎	
		具備系統嚴重錯誤之通知機制(例如電子郵件或簡訊)			◎	
	安全系統發展生命週期測試階段	執行「弱點掃描」安全檢測	◎	◎	◎	
		執行「滲透測試」安全檢測			◎	
	安全系統發展生命週期部署與維運階段	作業平台定期更新並關閉不必要服務及埠口(Port)	◎	◎	◎	
		針對系統依賴的外部元件或軟體，注意其安全漏洞通告，定期評估更新	◎	◎	◎	
		系統依賴的外部元件或軟體，不使用預設或空的密碼	◎	◎	◎	
	安全系統發展生命週期委外階段	資訊系統開發若委外服務應將系統發展生命週期各階段依安全等級將安全需求納入委外合約	◎	◎	◎	
	獲得程序	開發、測試以及正式作業環境應作區隔		◎	◎	
	資訊系統文件	應儲存與管理系統發展生命週期之相關文件	◎	◎	◎	

系統與通訊保護	傳輸之機密性與完整性	機敏資料傳輸時，採用加密機制		◎	◎	
		使用公開、國際機構驗證且未遭破解的演算法			◎	
	資料儲存之安全	參數設定或系統設定存放處，限制存取或進行適當保護			◎	
		機敏資料儲存時，採用加密機制			◎	
系統與資訊完整性	資訊系統監控	發現資訊系統有被入侵跡象時，應通報機關特定人員	◎	◎	◎	
		監控資訊系統，以偵測攻擊和未授權之連線，並識別資訊系統之未授權使用		◎	◎	
		資訊系統應採用自動化工具監控進出之通信流量，並於發現不尋常或未授權之活動時針對該事件進行分析			◎	
	軟體及資訊完整性	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		◎	◎	

資料來源： 本計畫整理