校園網頁應用程式安全之研究-以淡江大學為例

黃明達 博士
淡江大學資訊管理研究所
mdhwang @mail.tku.edu.tw

詹益璋
淡江大學資訊管理研究所 研究生
699630470@s99.tku.edu.tw

摘要

現今網頁應用程式應用更加複雜，產生漏洞可能性越大，本研究透過 IBM 的網頁應用程式弱點掃描軟體 AppScan，針對淡江大學學術單位一、二級與一級行政單位，進行網頁應用程式的掃描，研究目的有二，其一，透過弱點掃描之報告，找出淡江大學內網頁應用程式最主要的弱點。其二，針對淡江大學現存之網頁應用程式弱點，提出改善建議。

本研究採用 OWASP 2010 十大弱點與 AppScan 的弱點分類對比，發現目前淡江大學校內以 OWASP 排名前四名依序為注入弱點風險、遭破壞的鑑別與連線管理、跨站請求偽造與跨腳本攻擊四種弱點。若針對此四種弱點改善，將能有效改善校內網頁應用程式的安全。

關鍵詞：網頁應用程式弱點、OWASP、網頁應用程式安全

校園網頁應用程式安全之研究-以淡江大學為例

一、緒論

(一)研究背景與動機

隨著科技的發達，網路已經幾乎成為每個人生活中的必需品，99 年國內上網家戶（80.7%）及上網人口（70.9%）雙雙創新高，此外全臺 12 歲以上網路使用人口目前超過 1,446 萬人，較 98 年增加約 80 萬人。[7]再加上行動上網裝置的普及，現代人每天瀏覽網頁的時間也大幅度的增加，台灣網路資訊中心「2011 台灣無線網路使用調查報告」指出曾經上網的民眾「最近半年使用」過行動上網之電訪受訪者比例為 24.48%;「曾經使用」過行動上網之電話訪問受訪者。比例為 35.44%。另外行動網民有四成三是「每天使用」。[6]在這種環境之下，無論是政府機關、各級學校或者是私人企業，都已經把網頁的建置當作一項不可或缺的服務項目，網頁的功能也從以往的靜態網頁轉變為使用網頁應用程式提供動態功能的動態網頁。

網頁應用程式是指使用網頁瀏覽器在網路上操作的應用程式。[29]行政院國家資通安全會報又定義，泛指用於網頁應用服務的程式，架於網頁伺服器中，瀏覽器透過 URL(Uniform Resource Locator)網址與伺服器連結，並將網頁內容從伺服器端傳送至瀏覽器端，最後由瀏覽器呈現[9]。因目前網頁應用程式的普及，現今強調個人化功能及具備高度親和力、提供影音分享平台網站的 Web2.0 時代，如此不斷地進步，就是要將網路發展的應用更人性化，愈接近人性化的網頁應用程式其原始程式碼勢必更複雜，而複雜程式產生漏洞的可能性也就愈大，更使得有心人士能夠利用這些漏洞入侵網站。[12]此外行政院「2010 資通安全政策白皮書」也提到企業普遍建置防火牆以及入侵偵測系統，駭客欲從網路層入侵的難度大為升高。因此繞過防火牆與入侵偵測系統的阻擋，針對網頁應用程式的惡意攻擊行為日益增加。[8]

由於應用範圍日廣，網頁應用程式安全已經逐漸的受到重視，駭客們也悄悄的將焦點轉移到網頁應用程式開發時所會產生的弱點來進行攻擊與破壞。[3]校園網頁應用程式安全曾發生過 2008 年就傳出駭客盜取國中網站的個人資料，賣給補教業者牟利的案件。[5]2007 年也發生「學生扮駭客入侵學校網頁多放 3 天寒假」的等事件。[10]種種校園網頁應用程式安全事件層出不窮。因此網頁應用程式安全問題對學校就格外顯得特別重要。

以淡江大學為例，淡江大學目前幾乎所有系所與行政單位皆有所屬的網頁，而在網頁上除了提供一般性的圖文內容外，網頁上也有提供 EMAIL 的服務以及教學支援平台供老師發布訊息或學生上傳作業的使用，因此，在網頁的使用上也相當普遍，希望以弱點掃描的方式來對淡江大學內各級單位網頁應用程式作檢測，來了解是否存在網頁應用程式的弱點。

(二)研究目的

本研究透過淡江大學資訊中心現有的弱點掃描工具來做滲透測試，淡江大學各級機關中，都有網頁的應用，但因目前建置的狀況，以委託學生或老師幫忙建置網頁的單位占大多數，網頁沒有統一的標準規格，導致目前各單位網頁應用程式弱點數目參差不齊，又可能因建置人員更替，對於網頁應用程式安全管理難以掌握，因此本研究希望透過弱點掃描報告，來達到以下兩點目的：

1. 針對淡江大學學術單位的一、二級單位，以及一級行政單位，使用弱點掃描工具對網頁應用程式弱點作滲透測試，找出淡江大學內網頁應用程式最主要的弱點。

2. 期望各單位透過測試結果了解目前存在哪些網頁應用程式弱點，並且提出改善建議給各單位，讓各單位對網頁應用程式弱點有修正參考的依據。

(三)研究範圍

本研究所使用之弱點掃描工具，是以滲透測試的方式作掃描，因此在測試之時有使網頁應用程式服務中斷的疑慮，經淡江大學資訊處之資訊長同意，且淡江大學資訊處購有網頁應用程式弱點掃描工具 AppScan，經由資訊處同意與淡江大學網路組之協助才得以進行掃描，透過掃描結果對於高風險提出的改善建議，供淡江大學各單位之網頁應用程式弱點改善參考，或其他學校網頁應用程式弱點之參考範例。

淡水校園設有 8 個學院、35 學系。[13]各系所皆有提供網頁應用程式服務。本研究使用弱點掃描軟體，掃描範圍為學術單位一、二級以及一級行政單位，因二級行政單位之網頁許多皆包含於一級單位網頁內，所以只針對一級行政單位進行網頁測試，本研究所進行網頁應用程式弱點測試並且將弱點掃描軟體之弱點報告作歸納整理，分析淡江大學之網頁存在的弱點分布以及網頁建置人員或單位之資訊，探討淡江大學中網頁應用程式弱點存在之原因及提出修正建議。

(四)研究架構

本研究總共分為五章，第一章為緒論，陳述本研究之研究目的及背景，以及研究所涵蓋之範圍為何。第二章為文獻探討，就網頁應用程式弱點作定義，並介紹 OWASP TOP 10 2010 十大弱點的內容，及網頁應用程式所受到的威脅，並且對先前校園網頁應用程式研究的回顧。第三章為研究方法及過程，將本研究使用工具對網頁應用程式弱點作測試之研究過程描述。第四章為研究結果與分析，將 AppScan 所產出之弱點報告做歸納分析，並將 AppScan 所定義之弱點與 OWASP TOP 10 2010 十大弱點作對照，以及提出本研究主要的發現。第五章為結論與建議，將本研究之成果提出，並且對各單位網頁應用程式弱點提出改善建議，最後對於未來之研究提出建議。

圖 1-1 研究架構

二、文獻探討

(一)網頁應用程式弱點定義

弱點，主要是指一瑕疵導致功能無法完全呈現(即使在正確操作情況下)、洩漏系統資料或獲得未經信任授權的控制等，不符合操作預計情況的結果出現。[15]另外弱點也可說是一個可以被利用的漏洞。[25]維基百科定義弱點就是一個可能使威脅發生的漏洞。[28]因此網頁應用程式弱點可被解釋為，存在於網路應用程式中的瑕疵，可被攻擊者利用來竊取資訊或是不合法的取得網頁應用程式的控制權。

現今網頁應用程式的使用非常普遍，但是在開發網頁時，並不是每一個開發人員都是具有高度的資安意識，因此，開發者可能在建置網頁之初，沒有將網頁應用程式的安全性考量在內，只專注於網頁所能提供的服務內容，或者是開發者技術不良，導致網頁應用程式存在許多弱點，這些弱點可能使網頁應用程式本身及使用者造成威脅。

(二)OWASP 2010 十大弱點

OWASP(Open Web Application Security Project)為國際間非營利組織，長期對於網頁

應用程式之弱點提出警告及建議，此組織關注網頁應用程式弱點並整理 TOP 10 的網頁應用程式弱點公布，目前最新的版本為 OWASP 2010[24]，目前學校所使用的 IBM AppScan 的版本為 8.5.0.0 所產出的安全報告之一，就是使用 OWASP 2010 所提供的 TOP 10 分類。以下針對 OWASP 2010 之 TOP 10 安全問題作說明：

1. A1 注入攻擊(Injection)

攻擊者將攻擊字串注入正常的語法中，使得網站伺服器出現非預期的結果，並執行攻擊者輸入的指令。[23]注入攻擊的目的是攻擊者注入並執行指定的命令到易受攻擊的應用程式。[20]

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

跨站點腳本攻擊是一種注入的弱點，對良性和信任的網站注入惡意腳本。[21]若網頁沒有對上傳的內容進行過濾，則攻擊者將可透過攻擊語法插入動態網頁程式之中，導致使用者在瀏覽網頁的過程中，不自覺地執行攻擊語法。[23]

3. A3 遭破壞的鑑別與連線管理(Broken Authentication and Session Management)

遭破壞的鑑別與連線管理是網站應用程式中自行撰寫的身分驗證相關功能有缺陷。[14]「遭破壞的鑑別與連線管理」提供了攻擊者另外一條道路，網站的認證機制或連線管理不良，可能導致攻擊者非法取得網站部分或完整權限，在未正常登入的情況下對網站的功能進行操作，進而造成嚴重的後果。[23]

4. A4 不安全的直接物件參照(Insecure Direct Object Reference)

網站管理者有時可能因為貪圖方便或是疏忽會，將重要的檔案或目錄放在網頁伺服器上，雖然未公開，但是攻擊者還是有可能從其他管道取得檔案，造成資訊外洩。[23]

5. A5 跨站請求偽造(Cross Site Request Forgery)

跨站請求偽造是迫使使用者在他目前認證的網頁應用程式上執行不必要的行動。[22]「跨站請求偽造」這項攻擊是當攻擊者能夠取得並且偽造這些操作的語法，再透過搭配跨站腳本攻擊等方式，則可讓不知情的使用者在維持登入的情況下，執行攻擊者所偽造的語法，進而對網站伺服器執行攻擊者所安排的請求。[23]

6. A6 安全設定不當(Security Misconfiguration)

不安全的設定包含沒有定期進行必要更新、使用預設帳號密碼、使用預設檔案或目錄名稱、開啟非必要埠口或服務、沒有正確設定防火牆或 IDS 等。[23]

7. A7 不安全加密儲存(Insecure Cryptographic Storage)

使用較弱的加密演算機制或金鑰，甚至沒有加密，或將機敏資料存放至外部使用者容易取得的地方，甚至公開存放，則該網站就具有「不安全加密儲存」的弱點。[23]

8. A8 未限制 URL 存取(Failure to Restrict URL Access)

「未限制 URL 存取」攻擊者可透過修改 URL 方式，非法存取未被授權瀏覽之頁面。其他像是直接存取後台管理登入頁面，或是透過更改 URL 繞過認證機制而執行某些特定功能，都屬於此弱點的影響。[23]

9. A9 不足的傳輸層保護(Insufficient Transport Layer Protection)

「不足的傳輸層保護」很有可能會讓機敏資料曝露在遠方的攻擊者面前。因此為了保護這些資料不被竊取，網站應該要求使用者在傳送機敏資料時，務必要使用一定強度的加密機制傳送(如 SSL)，避免資料外洩的風險。[23]

10. A10 未驗證的重導和轉送(Unvalidated Redirects and Forwards)

若網站具有「未驗證的重導和轉送」這項弱點，攻擊者可利用重導或轉送這項功能，透過偽造 URL 的方式，將使用者在不知情的情況下，連結至惡意網站或釣魚網站，造成使用者嚴重的損失。[23]

(三)網頁應用程式之威脅

WASC(Web Application Security Consortium)是由許多研究資安的專業人員以及一些公司組成專門研究網站安全的團體，其中一項專案就是將網站安全的威脅進行分類。[4] 在 AppScan 當中也採用 WASC 的威脅分類，分類如表 2-1。

表 2-1WASC 威脅分類

威脅名稱			
強制入侵	跨網站 Scripting	XPath 注入	分割 HTTP 回應
鑑別不足	緩衝區溢位	目錄檢索	空值位元組注入
認證/階段作業預測	格式字串	資訊洩漏	XML 實體展開
授權不足	LDAP 注入	路徑遍訪	應用程式品質測試
不安全製作索引	OS 接管	可預測的資源位置	併入遠端檔案
階段作業固定	SQL 注入	濫用功能	偽造跨網站要求
URl 重新導向程式濫用	SSI 注入	濫用 SOAP 陣列	未充分設定階段作業有效期限
XML 屬性爆炸	阻斷服務	XML 外部實體	應用程式隱私測試
內容盜用			

其中最需要注意的為 SQL 注入與跨網站 Scripting。SQL 注入的產生是當攻擊者能夠輸入一系列的 SQL 語句通過“查詢”操縱數據輸入到應用程序中的。[17] 一個成功的 SQL 注入攻擊，最嚴重可導致資料遺失或損壞，拒絕訪問，成功的攻擊是不需要取得帳戶權限來達到一個完整的主機接管。[19]

跨網站 Scripting 攻擊中，將網頁應用程式的腳本代碼注入到一個網頁應用程式，然後將其發送到用戶的網頁瀏覽器。在瀏覽器中，此腳本代碼被執行，用來傳輸敏感數據到第三方（即攻擊者）。[18] 跨網站 Scriptin 最主要的產生原因，是網頁開發者未將使用者的輸入做合適的處理，就呈現在網頁上。於是惡意的 JavaScript 程式碼就會在使用者瀏覽網頁時被執行。[16] 跨網站 Scriptin 攻擊嚴重時，攻擊者可以盜用使用者的 cookie 來冒充使用者。Cookie 是指瀏覽器使用時儲存在使用者端的文件，如：登入資訊等訊息。[27] 另外 Hung-Jui Ke 等提出：被送到伺服器前過濾數據、使用應用程式防火牆、加強網站管理等方式來阻擋 XSS 攻擊。[26]

(四)校園網頁應用程式弱點研究

蔡震天(2010)與姚依君(2010)皆以網頁應用程式弱點的主題，對淡江大學內 6 個學院 31 個學系以及教務處的總共 38 個網頁應用程式作掃描，掃描結果之弱點排序如表 2-2。

蔡震天所使用的弱點分類是以 OWASP TOP 10 2007 為標準，使用 Fortify 網頁原始碼檢測工具，實際的針對弱點產生的原始碼位置偵測，並提供修改的方法，並也利用弱點掃描工具作後續追蹤。蔡震天的研究指出，淡江大學的網頁資料庫的應用較少，因此排名出來的 OWASP 的排名注入弱點排名落到第三名，與企業之排名有所差異。[15]

姚依君是透過 AppScan 對網頁應用程式作掃描，歸納弱點並比較 OWASP TOP 10 2007 排名校園內與企業排名的差異，另外彙整攻擊手法提供網頁維護者作修補參考之用。姚依君另外指出，OWASP 的校園內排名與企業界之排名不同，可能的原因在於網頁提供的服務性質的差異所造成。淡江大學之網頁應用程式多為公告訊息之靜態網頁服務，而業界之網頁應用程式除提供訊息外，為提高互動性，提供用戶端與後端資料庫存取管道，故所存在之弱點類型排序亦相異。[11]

表 2-2 姚依君與蔡震天研究之 OWASP TOP 10 2007 排名之比較

姚依君(2010)	蔡震天(2010)
(A6)Information Leakage and Improper Error	(A1)Cross Site Scripting (XSS)

Handling	
(A4)Insecure Direct Object Reference	(A5)Cross Site Request Forgery (CSRF)
(A10)Failure to Restrict URL Access	(A2)Injection Flaws
(A7)Broken Authentication and Session Management	(A6)Information Leakage and Improper Error Handling
(A3)Malicious File Execution	(A8)Insecure Cryptographic Storage
(A8)Insecure Cryptographic Storage	(A4)Insecure Direct Object Reference
(A1)Cross Site Scripting (XSS)	
(A9)Insecure Communications	
(A5)Cross Site Request Forgery (CSRF)	
(A2)Injection Flaws	

OWASP Top 10 2007 年版的評比方式非常簡化，其以 2006 年的 MITRE 弱點統計趨勢（Vulnerability Trends）為參考，找出與網頁應用程式相關的前十大安全問題，但弱點發生次數頻繁卻不表示其風險就高。[2]因此本研究採用的 OWASP Top 10 2010 版本風險項目就有更動。

以上兩研究之範圍都是淡江大學資訊處網路管理組，所統一管理的網頁應用程式，本研究的範圍延伸到其他非網路管理組管理之網頁應用程式，期望經由掃描範圍的擴大，比較結果後，找出淡江大學之網頁應用程式弱點排名與先前研究是否有不同。並藉由針對弱點風險的分類，來找出需要優先修改的高風險提出改善建議。

三、研究方法與過程

(一)研究對象

本研究的對象為淡江大學內學術單位一、二級共 42 個單位，以及一級行政單位共 9 個單位之網頁應用程式，總共 51 個網站作弱點滲透測試，自 2011 年 10 月起至 12 月底止。因淡江大學之學術單位中，屬於二級單位的系所皆有各自獨立的網頁，行政單位之二級單位網頁，大部分皆附屬於一級單位之下，因此藉由學術單位一、二級與一級行政單位之網頁應用程式弱點掃描，來檢視淡江大學網頁應用程式弱點的概況。

目前淡江大學資訊處委託委外廠商規畫建置標準樣版網頁，本研究主要是針對目前尚未使用資訊處所提供之標準樣版的網頁作弱點測試。雖淡江大學內網頁應用程式風險以低風險與參考風險數目較多，但各校之網頁應用程式建置生態均類似，多為學生或老師所建置，因此以淡江大學為例針對高風險所做之分析建議，亦可提供各單位及各校相關人員作參考。

(二)研究工具

本研究使用網頁應用程式掃描軟體 IBM Rational AppScan 8.5.0.0 版本：AppScan 是一套自動化弱點掃描工具，用來檢測網頁應用程式的安全性，找出系統的資安漏洞，並一一提供詳盡的處理建議。[1]AppScan 是以模擬的攻擊手法對網頁應用程式作弱點測試，測試之初會先對網站內之 URL 數目作探測，並計算出對該網站需要進行多少次滲透測試，並將測試結果分高、中、低、參考資訊四級風險呈現，並統計各級風險之數目，並顯示各級風險之項下弱點名稱。例如：高風險：SQL 注入、跨網站 Scripting 等項目。掃描完成後可利用 AppScan 客製化產出報告，以勾選方式選擇產出報告內所需之項目，例如：安全問題、補救作業等項目。

因 AppScan 之網頁應用程式弱點偵測，是模擬實際攻擊網頁應用程式的滲透測試，在使用時需特別注意是否會因掃描軟體執行導致網頁應用程式服務中斷，對於有建置防火牆之網頁，也可能因防火牆啟動使得測試無法順利進行。

(三)問卷設計

本研究問卷設計之目的在於，了解各單位目前使用之網頁建置單位或人員，並且將

目前網頁應用程式之弱點讓各單位有概略的認識，透過問卷訪談得知各單位對目前使用的網頁以及現存已知弱點數目滿意程度作瞭解。此外期望透過建置方式的資訊，分析其網頁應用程式之建置單位或人員，是否對目前網頁應用程式存在的弱點有影響，另外，透過問卷詢問各單位是否願意透過標準樣版網頁之使用，來達到統一管理網頁應用程式風險的目標，以降低網頁應用程式弱點可能帶來的威脅與影響，讓各單位有其他建置網頁應用程式的選擇。

(四)研究方法與過程

本研究使用 AppScan 選定淡江大學學術單位一、二級，以及一級行政單位之網頁應用程式進行滲透測試，再以問卷方式詢問各單位的建置方式並將各單位目前存在的網頁應用程式弱點彙整，並將統計結果歸納分析，找出目前所存在高風險的網頁應用程式主要高風險弱點為何？分析弱點發生的可能原因，再針對主要的高風險弱點威脅提出改善建議，讓各單位能對其網頁應用程式作改善。

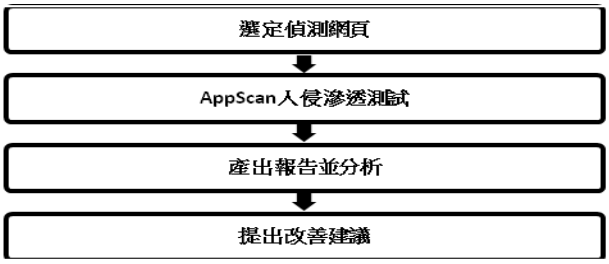

圖 3-1 研究方法與過程

四、研究結果與分析

(一)校園網頁應用程式之弱點分析

本研究對淡江大學學術單位一、二級，以及一級行政單位作滲透測試發現，以AppScan之弱點分類顯示(表4-1)，目前已知的網頁應用程式風險當中總數為：12,157個，其中高風險數為：1,618個；中風險為：623個，低風險為：5,231個，參考資訊為：4,685個。

以影響程度最為嚴重的高風險來說,對照WASC威脅風險項目可以發現,其中以SQL注入與跨網站 Scripting兩大風險數目佔了高風險總數高達81%。其餘高風險數僅佔19%，顯示目前淡江大學內之網頁應用程式威脅最高的就是此兩種類型之弱點，因此整體而言，如果各單位皆對此兩種風險做防堵，便可大大降低淡江大學內網頁應用程式之弱點威脅。

表4-1淡江大學風險個數表

風險等級	高風險	中風險	低風險	參考風險	總和
風險個數	1,618	623	5,231	4,685	12,157
百分比	13.31%	5.15%	43.03%	38.45%	100.00%

表4-2 高風險數量百分比

WASC 之威脅名稱	SQL 注入	跨網站 Scripting	高風險
個數	731	583	1,618
百分比	45.17%	36.03%	100%

本研究重點將放在高風險類型之弱點做分析，高風險可能讓攻擊者輕易取得管理者權限導致系統中斷，或者洩漏網頁內敏感資訊等危險，對於網站會有立即性的危害，因此以個別系所或行政單位來分析，高風險數量前三名與一級單位高風險數目前三名如表4-3。這些單位之高風險數目除覺生紀念圖館外，皆有超過一百筆的高風險

存在，顯示這些單位對於網頁應用程式弱點管理是需要加強的，需要儘快將網頁應用程式之高風險弱點告知網頁建置或維護人員將其排除，以保護網頁應用程式的安全。

表4-3高風險量前三名

高風險數量前三名		一級單位高風險數量前三名	
單位名稱	數目	單位名稱	數目
資訊圖書學系	477	秘書處	200
秘書處	200	蘭陽校園主任室	143
商管聯合碩士在職專班	181	覺生紀念圖書館	22

此外，透過問卷發現，目前淡江大學各單位所建置的網頁應用程式，建置的方式最多是由老師或學生所負責建置，也有資訊處 數位設計組及資管網路策進會建置，其他還有外包廠商或是由職員共同建置等方式完成(表 4-4)，透過交叉分析來分析建置單位或人員，對於網頁應用程式弱點的高風險數量之間的影響，經過列聯相關分析的結果，建置單位與高風險數目之相關系數為 0.144 ，顯著性 p 值為 0.985>0.05，未達顯著水準。卡方值為 1.011，p 值為 0.985>0.05，未達顯著水準。顯示由兩者之間相關的程度並不高。發現建置單位或人員對於高風險弱點之產生並無顯著相關，表示不論是由誰來建置網頁應用程式，其對於高風險弱點的多寡並無關聯。

表 4-4 各單位網頁建置情況

網頁建置單位/人員	資訊處 數位設計組	老師/學生	資管網路策進會	其他	總和
個數	3	30	1	14	48
百分比	6.52%	65.21%	2.17%	30.43%	100.00%

(二)OWASP 十大弱點與 AppScan 弱點對照

現在世界上 OWASP 之標準為大多數的機關單位認可並採用，表 4-5 為 OWASP TOP 10 2010 之代碼與弱點名稱，因此在此也將 AppScan 內的風險分類與 OWASP TOP 10 2010 的風險作對照(表 4-6)。

表 4-5 OWASP TOP 10 2010 名稱表

代碼	弱點名稱
A1	注入弱點風險(Injection)
A2	跨腳本攻擊(Cross site scripting) (XSS)
A3	遭破壞的鑑別與連線管理(Broken authentication and session management)
A4	不安全的直接物件參考(Insecure direct object reference)
A5	跨站請求偽造(Cross site request forgery) (CSRF)
A6	錯誤或不安全的系統組態(Security Misconfiguration)
A7	不安全的加密儲存方式(Insecure cryptographic storage)
A8	網址存取控制失當(Failure to restrict URL access)
A9	傳輸層保護不足(Insufficient transport layer protection)
A10	未驗證的轉址與轉送(Unvalidated Redirects and Forwards)

表 4-6 OWASP TOP 10 2010 與 AppScan 對照表

代碼	AppScan 分類表
A1	找到資料庫錯誤型樣、盲目的 SQL 注入、SQL 注入、盲目的 SQL 注入(時間型)、使用 DECLARE、CAST 和 EXEC 的 SQL 注入、參數值溢位、IMAP MX 注入、POP3 MX 注入
A2	跨網站 Scripting、DOM 型跨網站 Scripting

A3	跨網站 Scripting、未加密的登入要求、直接存取管理頁面、鏈結注入（有助於偽造跨網站要求）、Flash 參數 Allow Script Access 設為 always、啟用 TRACE 與 TRACK HTTP 方法、在階段作業 Cookie 中遺漏 Htt pOnly 屬性、HTTP 回應分割、DOM 型跨網站 Scripting、未更新階段作業 ID
A4	Microsoft FrontPage '_vti_cnf' 資訊洩漏、下載暫存檔、Apache 多重視圖攻擊、偵測到隱藏目錄、找到可能的伺服器路徑揭露型樣、找到目錄清單型樣、HTML 註解機密性資訊揭露、目錄清單、Microsoft FrontPage 目錄清單、找到 Web 應用程式原始碼揭露型樣、找到壓縮的目錄、Robots.txt 檔網站結構曝光、Microsoft IIS 'Translate: f' 原始碼揭露、偵測到應用程式測試 Script
A5	跨網站 Scripting、未加密的登入要求、直接存取管理頁面、鏈結注入（有助於偽造跨網站要求）、Flash 參數 Allow Script Access 設為 always、在階段作業 Cookie 中遺漏 HttpOnly 屬性、HTTP 回應分割、DOM 型跨網站 Scripting、啟用 TRACE 與 TRACK HTTP 方法、未更新階段作業 ID
A6	找到電子郵件位址型樣、Microsoft FrontPage '_vti_cnf' 資訊洩漏、偵測到檔案的替代版本、下載暫存檔、找到可能的伺服器路徑揭露型樣、找到目錄清單型樣、目錄清單、Microsoft FrontPage 目錄清單、找到 Web 應用程式原始碼揭露型樣、浮點數值阻斷服務、Flash 參數 AllowScriptAccess 設為 always、找到壓縮的目錄、未加密的 __VIEWSTATE 參數、偵測到應用程式測試 Script、未更新階段作業 ID
A7	找到電子郵件位址型樣、偵測到檔案的替代版本、HTML 註解機密性資訊揭露、未加密的登入要求、未加密的 __VIEWSTATE 參數
A8	找到電子郵件位址型樣、Microsoft FrontPage '_vti_cnf' 資訊洩漏、下載暫存檔、Apache 多重視圖攻擊、偵測到隱藏目錄、找到可能的伺服器路徑揭露型樣、找到目錄清單型樣、目錄清單、直接存取管理頁面、Microsoft FrontPage 目錄清單、鏈結注入(有助於偽造跨網站要求)、找到 Web 應用程式原始碼揭露型樣、Flash 參數 Allow Script Access 設為 always、找到壓縮的目錄、Robots.txt 檔網站結構曝光、Microsoft IIS 'Translate: f' 原始碼揭露、未加密的 __VIEWSTATE 參數、偵測到應用程式測試 Script、啟用 TRACE 與 TRACK HTTP 方法、未更新階段作業 ID
A9	未加密的登入要求
A10	—

淡江大學 OWASP TOP 10 2010 之排名，前三名分別為：(A8)網址存取控制失當(Failure to restrict URL access)、(A6)錯誤或不安全的系統組態(Security Misconfiguration)、(A7)不安全的加密儲存方式(Insecure cryptographic storage)。以此三種風險對照 AppScan 中弱點的分類顯示，是主要以中低風險及參考風險所組成，中、低風險等級以下之弱點，對於改善網頁應用程式之威脅較無急迫性，因此本研究採取只以高風險弱點對照 OWASP TOP 10 2010 排名(表 4-7)排名前三名為：(A1)注入弱點風險(Injection)、(A3)遭破壞的鑑別與連線管理(Broken authentication and session management)、(A5)跨站請求偽造(Cross site request forgery) (CSRF)三種風險。

其中原本排名第一名的 A8 弱點與 A4 弱點，在剔除掉中低風險及參考資訊風險弱點之後，風險個數變為 0，第二名的 A6 風險個數也大幅降低，此現象表示如未作風險篩選直接以 OWASP TOP 10 2010 對淡江大學內的網頁應用程式弱點作排名，前三名風險個數雖多，但是都是中風險等級以下的弱點所組成，對於改善網頁應用程式弱點並無立即的急迫性，因此以未篩選過之風險排名對網頁應用程式作改善，無法有效改善風險最高之弱點。

表 4-7 淡江大學 OWASP TOP 10 2010 排名

淡江大學 OWASP TOP 10 2010 排名(高風險)	
代碼	OWASP 弱點名稱
A1	注入弱點風險(Injection)
A3	遭破壞的鑑別與連線管理(Broken authentication and session management)
A5	跨站請求偽造(Cross site request forgery) (CSRF)
A2	跨腳本攻擊(Cross site scripting) (XSS)
A7	不安全的加密儲存方式(Insecure cryptographic storage)
A9	傳輸層保護不足(Insufficient transport layer protection)
A6	錯誤或不安全的系統組態(Security Misconfiguration)
A4	不安全的直接物件參考(Insecure direct object reference)
A8	網址存取控制失當(Failure to restrict URL access)
A10	未驗證的轉址與轉送(UnvalidatedRedirects and Forwards)

(三)各學院與行政單位弱點分析

本研究也針對各學術單位以及行政單位之弱點狀況做分析，各學術單位與行政單位之高風險對照 OWASP TOP 10 2010 排名如表 4-8。

表 4-8OWASP TOP 10 2010 各單位風險數量排名

排名	1	2	3	4	5	6	7	8	9	10
文學院	A1(367)	A3(195)	A5(195)	A2(126)	A7(69)	A9(69)	A6(1)	A4(0)	A8(0)	A10(0)
理學院	A3(10)	A5(10)	A7(10)	A9(10)	A1(9)	A2(0)	A4(0)	A6(0)	A8(0)	A10(0)
工學院	A3(11)	A5(11)	A2(10)	A1(8)	A6(5)	A7(1)	A9(1)	A4(0)	A8(0)	A10(0)
商學院	A1(62)	A3(35)	A5(35)	A2(28)	A7(7)	A9(7)	A6(1)	A4(0)	A8(0)	A10(0)
管理學院	A3(155)	A5(155)	A1(152)	A2(130)	A7(23)	A9(23)	A6(3)	A4(0)	A8(0)	A10(0)
外語學院	-	-	-	-	-	-	-	-	-	-
國際研究學院	A3(4)	A5(4)	A2(3)	A7(1)	A9(1)	A1(0)	A4(0)	A6(0)	A8(0)	A10(0)
教育學院	A3(40)	A5(40)	A1(39)	A2(34)	A6(14)	A7(6)	A9(6)	A4(0)	A8(0)	A10(0)
全球創業發展學院	A1(75)	A6(18)	A3(17)	A5(17)	A2(9)	A7(8)	A9(8)	A4(0)	A8(0)	A10(0)
其他學術單位	A1(14)	A3(9)	A5(9)	A7(5)	A9(5)	A2(4)	A6(1)	A4(0)	A8(0)	A10(0)
行政單位	A3(248)	A5(248)	A2(238)	A1(116)	A6(12)	A7(10)	A9(10)	A4(0)	A8(0)	A10(0)

觀察排名後發現除少數單位外，各單位風險排名之前五名皆包含 A1、A3、A5、A2，此四種風險分類最主要是以 SQL 注入與跨網站 Scripting 為的兩種弱點所組成。

此外各單位數量與其高風險總數透過回歸分析發現，相關系數為 0.192，顯著性 p 值為 0.206>0.05，未達顯著水準，顯示單位的數量與其高風險數量，並無顯著相關，高風險的數量，並未因單位數量較多而影響。

(四)主要研究發現

在本章中，本研究以 AppScan 的弱點分類與 OWASP TOP 10 2010 之弱點分類去分析，發現不論是以整體而言，或是將各學術單位與行政單位做分類比較，顯示注入弱點風險(Injection)、遭破壞的鑑別與連線管理(Broken authentication and session

management)、跨站請求偽造(Cross site request forgery) (CSRF)與跨腳本攻擊(Cross site scripting)是淡江大學目前網頁應用程式的最主要弱點。

雖然各單位的建置方式各異，但所產生的網頁應用程式弱點卻都極為相似，因此不論是由何人所開發之網頁應用程式，若針對這四項網頁應用程式弱點作改善，就能防堵大部分的高風險弱點。此外注入弱點風險(Injection)類型中最需要注意的是 SQL 注入風險。跨站點腳本攻擊(Cross site scripting)、遭破壞的鑑別與連線管理(Broken authentication and session management)、跨站請求偽造(Cross site request forgery)此三類弱點皆以跨網站Scripting(XSS)為主要組成風險。

不論 SQL 注入或是跨網站 Scripting 的弱點，雖然造成威脅方式不同，但其AppScan 提供的改善方式皆為：「確認使用者輸入未包含危險的字元，如：「;」(分號)、「%」(百分比符號)、「'」(單引號)、「"」(引號) 等，便可能防止惡意的使用者讓您的應用程式執行非預期的作業，例如：啟動任意 SQL 查詢、內嵌執行於用戶端的JavaScript 程式碼、執行各種作業系統指令等等。」。

遭破壞的鑑別與連線管理與跨站請求偽造風險中，還有「未加密的登入要求」這項主要弱點，未加密的登入要求是指機密性輸入欄位（如：使用者名稱、密碼和信用卡號碼）傳遞時未加密。此風險的AppScan改善建議為：傳送機密性資訊時，一律使用 SSL 和 POST（主體）參數。

對照姚依君(2010)之OWASP排名是未排除中、低以下風險所整理，因此以高風險組成為主的注入風險與跨網站scripting等弱點排名會落到後五名，有可能被忽略或是較晚才改善。在蔡震天(2010)的研究中認為因學校網頁應用程式資料庫應用較少，因此注入風險排名掉至第三名，但本研究將掃描之網頁應用程式擴大到51個之後，發現淡江大學內注入風險最多。推測淡江大學網頁應用程式服務資料庫使用，在近幾年已經增加或是因先前研究對象環境統一導致結果有所出入。參照下面比較表4-9。

表4-9 本研究與姚依君、蔡震天研究比較

本研究	姚依君(2010)	蔡震天(2010)
A1注入弱點風險(Injection)	(A6)Information Leakage and Improper Error Handling	(A1)CrossSite Scripting (XSS)
A3遭破壞的鑑別與連線管理(Broken authentication and session management)	(A4)Insecure Direct Object Reference	(A5)Cross Site Request Forgery (CSRF)
A5跨站請求偽造(Cross site request forgery) (CSRF)	(A10)Failure to Restrict URL Access	(A2)Injection Flaws
A2跨腳本攻擊(Cross site scripting) (XSS)	(A7)Broken Authentication and Session Management	(A6)Information Leakage and Improper Error Handling
A7不安全的加密儲存方式(Insecure cryptographic storage)	(A3)Malicious File Execution	(A8)Insecure Cryptographic Storage
A9傳輸層保護不足(Insufficient transport layer protection)	(A8)Insecure Cryptographic Storage	(A4)Insecure Direct Object Reference
A6錯誤或不安全的系統組態(Security Misconfiguration)	(A1)CrossSite Scripting (XSS)	
A4 不安全的直接物件參考(Insecure direct object reference)	(A9)Insecure Communications	
A8 網址存取控制失當(Failure to	(A5)Cross Site Request	

restrict URL access)	Forgery (CSRF)	
A10 未驗證的轉址與轉送 (UnvalidatedRedirects and Forwards)	(A2)Injection Flaws	

此外本研究另外對淡江大學資訊處委外廠商規畫建置的標準樣版網頁，挑選其中兩個網頁應用程式作樣本，同樣以 AppScan 作弱點滲透測試發現，透過資訊處委外廠商統一對標準樣版網頁之網頁應用程式弱點作改善後，高風險弱點數目已經有明顯的降低(表 4-10)，顯示由統一的管理單位對網頁應用程式管理，能夠一次性的改善多個網頁應用程式的風險，對於網頁應用程式的管理上，相較於各單位各自管理，能有效的降低所存在之風險，但標準樣版網頁之建置需要經費的支持，若不考慮成本限制，能統一使用標準樣版網頁，對於網頁應用程式弱點，更能有效的改善及管理。

表 4-10 標準樣版網頁高風險改善表

改善前		改善後	
單位名稱	高風險數目	單位名稱	高風險數目
歷史學系	1,281	歷史學系	155
日本語文學系	707	日本語文學系	83

五、結論

(一)結論

本研究透過淡江大學的掃描結果，除了提供淡江大學各單位作網頁應用程式弱點的修正參考，也提供給各學校資安人員及網頁應用程式開發或相關人員，作網頁應用程式弱點的修正參考，淡江大學目前不論是學術單位或是行政單位，皆使用網頁應用程式提供資訊與服務，也因此一旦網頁應用程式遭受攻擊而停擺，會讓使用者與網頁所屬單位造成有形或是無形的損失，本研究的結果：

1. 淡江大學高風險弱點數量前三名分別為資訊圖書學系、秘書處、商管聯合碩士在職專班。一級單位高風險數量前三名分別為秘書處、蘭陽校園主任室、覺生紀念圖書館。這些單位目前存在許多高風險弱點，高風險弱點容易受到攻擊者利用，弱點雖然不會主動造成網頁應用程式損壞，但還是需要儘早修正，以維護網頁應用程式的安全性。

2. 對網頁應用程式造成直接威脅的高風險，在淡江大學中主要依序為注入(Injection)弱點風險、遭破壞的鑑別與連線管理(Broken authentication and session management)、跨站請求偽造(Cross site request forgery) (CSRF)與跨腳本攻擊(Cross site scripting)四種弱點普遍存在網頁應用程式當中，顯示以目前淡江大學的網頁應用程式而言，是最先需要改善的四個風險問題。

3. 淡江大學之OWASP排名，經由排除中低等級以下風險弱點之後，排名順序更動頗大，顯示因中低等級以下風險弱點數量過多，有可能掩蓋高風險存在的狀況，因此在對網站應用程式弱點作改善時，須先過濾威脅程度較低之風險，針對高風險弱點優先作修正。

(二)改善建議

以下針對本研究此次所得到之結果，對淡江大學內之網頁應用程式提出改善建議：

1. 本研究透過問卷，將各單位網頁用程式弱點清單交給各單位，各單位瞭解目前網頁應用程式弱點概況後，應儘早將風險清單交與網頁開發人員，在最短的時間內將現存的高風險弱點作改善，以防止網頁遭受攻擊或入侵，導致網頁應用程式中可能存在的個資或其他資訊遭竊。

2. 經本研究測試發現，由淡江大學資訊處所規畫建置的標準樣版網頁，透過改善後高風險弱點數目明顯降低，若未來在經費許可的情況之下，能統一使用標準樣版網頁。

透過共同的樣式與管理單位，在有弱點產生的情況下，便可統一管理並改善，以節省管理網頁應用程式的時間與人力資源。

(三)未來研究建議

本研究因研究範圍限制，僅能以淡江大學內各單位之網頁應用程式作滲透測試，未能考慮到不同學校或是不同性質的單位所存在之網頁應用程式差異，因此所作的網頁應用程式弱點分析之結論不一定適用於各種不同組織之網頁應用程式。因此在此提出兩點建議，以供未來研究之參考：

1. 未來研究可加入各級學校或是企業等不同性質之組織，作網頁應用程式弱點滲透測試，其所得到網頁應用程式弱點之呈現會更加完整，並且以此結果所做的分析更廣泛使用於各組織中。

2. 本研究僅使用IBM的弱點掃描軟體AppScan來做測試，未來研究也可加入其他網頁應用程式弱點偵測工具，比較測試結果與建議，可以藉由各種網頁應用程式測試工具更完善的修正網頁應用程式之弱點。

參考文獻

[1]IBM- Web 應用程式零漏洞方案-Taiwan，http://www-01.ibm.com/software/tw/promotion/rational/appscan119.html，上網日期 2011 年 11 月 20 日。

[2] Jack Wu, 0WASP Top 10 2010 版初探網站資安風險管理- 如何透過源碼檢測與網站應用系統防火牆控制風險等級，http://armorize-cht.blogspot.com/2009/11/owasp-top10-2010.html，上網日期 2012 年 1 月 3 日。

[3] Taiwan-OWASP, https://www.owasp.org/index.php/Taiwan，上網日期 2011 年 10 月 17 日。

[4]TWNIC@NTUST網路應用安全知識庫，伺服端安全:3-4檢測與分析方法:最新弱點公布網站，http://knowledge.twisc.ntust.edu.tw/doku.php?id=3伺服端安全:3-4檢測與分析方法:最新弱點公布網站，上網日期 2011年12月27日。

[5]大紀元，http://www.epochtimes.com/b5/8/6/25/n2168156.htm，上網日期 2012年2月1日。

[6]台灣網路資訊中心(TWNIC)，<2011 台灣無線網路使用調查報告出爐>，2011年10月。

[7]行政院研究發展考核委員會，<99 年個人家戶數位落差調查報告>，中華民國九十九年十一月。

[8]行政院科技顧問組，< 2010 資通安全政策白皮書>，2010 年。

[9]行政院國家資通安全會報技術服務中心，<97 年度 Web 應用程式安全參考指引 V.2>，網址：http://www.giscc.org.tw/downloadFile.php?dispatch=download&sn=108，上網日期：2009 年 10 月 2 號。

[10]東海數位學堂， http://blog.yam.com/taso_tkb/article/35070350，上網日期 2012 年 2 月 1 日。

[11]姚依君，《網頁應用程式攻擊之研究-以淡江大學為例》資訊管理學系暨研究所，碩士論文，2010年。

[12]張智翔，中央研究院計算中心，通訊電子報，《淺談網路應用程式安全（一）》，http://newsletter.ascc.sinica.edu.tw/news/read_news.php?nid=1288，上網日期 2011年11月23日。

[13]淡江大學，<淡水校園簡介>，網址：http://foreign.tku.edu.tw/chinese/campus-tamsui.asp，上網日期：2011 年 11 月 19 號。

[14]翁浩正，中央研究院計算中心，通訊電子報，《淺談網路應用程式安全（二）》，http://newsletter.ascc.sinica.edu.tw/news/read_news.php?nid=1917，上網日期 2011年11月23日。
[15]蔡震天，《網頁應用程式原始碼弱點分析之研究—以淡江大學為例》資訊管理學系暨研究所，碩士論文，2010年。
[16]增加網路應用程式安全性 Improve web application security，http://brooky.cc/2011/05/18/improve-web-application-security/，上網日期 2012 年 1 月 16 日。
[17] Chris Anley ,Advanced SQL Injection In SQL Server Applications., http://www.ngssoftware.com, accessed 2012/1/3
[18] Philipp, Vogt., Florian, Nentwich., Nenad, Jovanovic.,Engin, Kirda., Christopher, Kruegel., and Giovanni, Vigna. "Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis", in In Proceedings of 14th Annual Network and Distributed System Security Symposium (NDSS 2007), 2007.
[19]Kate Riley, IST–Infrastructure Services, Protect your web applications from common threats, http://inews.berkeley.edu/articles/Aug-Sep2010/web-app-vulnerabilities,accessed 2012/1/17
[20]OWASP, Command Injection, https://www.owasp.org/index.php/Command_Injection,accessed 2011/12/18.
[21]OWASP, Cross-Site Scripting,https://www.owasp.org/index.php/Cross-site_Scripting_(XSS), accessed 2011/12/26.
[22]OWASP, Cross-Site Request Forgery (CSRF),https://www.owasp.org/index.php/CSRF, accessed 2011/12/26.
[23]OWASP TOP10-2010,http://www.owasp.org/index.php/Top_10,accessed 2011/12/5.
[24]OWASP Top 10 (2010 release candidate 1), http://www.slideshare.net/jeremiahgrossman/owasp-top-10-2010-release-candidate, accessed 2011/12/12.
[25]Practical Web Application Vulnerability Assessment, http://www.michaelboman.org/books/practical-web-application-vulnerability-assessment, accessed 2012/1/15.
[26]Shiuh-Jeng Wang.,Yao-Han Chang., Hung-Jui Ke., Wen-Shenq Juang .,"Digital Evidence Seizure in Network Intrusions against Cyber-crime on Internet Systems,"Journal of Computers Vol.18, No. 4, pp.69-78, 2008.
[27]Steven, Cook., "A Web Developer's Guide to Cross-Site Scripting", SANS Institute 2003, January 11, 2003.
[28]Wikipedia,Application Security., http://en.wikipedia.org/wiki/Application_security, accessed 2012/1/17.
[29]Wikipedia,Web Application., http://en.wikipedia.org/wiki/Web_application, accessed 2012/1/17.

A study of Campus Web Application Security－a case study of Tamkang University

Hwang, Ming-dar
Department of Information Management, Tamkang University
mdhwang @mail.tku.edu.tw

Chan Yi-chang
Department of Information Management, Tamkang University
699630470@s99.tku.edu.tw

Abstract

The web applications today use more complex, resulting the safety problem in a greater danger. In this study, by IBM's web application vulnerability detection software AppScan, we aim at the Tamkang University primary and secondary academic unit and primary administrative unit to do our web application scanning. There are two purposes in this study. First, by the report of the vulnerability detection, we can identify the major weakness of web applications in Tamkang University. Second, improve the existing weakness of web applications in Tamkang University.

In this study, we contrast the OWASP 2010 Top Ten weakness with AppScan's vulnerability classification. Found Tamkang University campus to the current the OWASP rank the top four in order to inject, Broken authentication and session management, Cross site request forgery and Cross site scripting four weaknesses.For this four weaknesses to improve, we can effectively improve the safety of the campus web applications.

Keywords: Web application vulnerabilities ,OWASP, Web application security