資安事件與 SSDLC 觀點探討資安

應用系統面臨多元資安風險之教戰手則（上）

撰文 | 叡揚資訊 資訊安全事業處 專案支援經理 鄭鈞元

現今資安事件層出不窮，舉凡國內外企業運作模式，大多在強調縱深防禦，投資於應用程式防火牆、IDS/IPS，多種硬體層次的防護措施，阻絕疑似惡意攻擊，但其實大多問題源於應用程式本身的缺陷，惡意訊息通常透過標準的 80 或 443 通訊端口發送請求，躲避特徵行為偵測，因此很難藉設備防護阻絕，應用程式本身安全的強化即成為確保系統安全重要的一環。

日常生活中的資安風險

「害人之心不可有、防人之心不可無。」物聯化時代的來臨意味著我們正逐漸無意識地交出個人隱私資訊給第三方資料中介或開發商，這些資料通常會去進行商業價值的分析及統計，但也有可能會被犯罪集團利用。因此，如何保護個人資料成了重要課題。而新一代的駭客更進一步地透過分工細密的組織去運作。

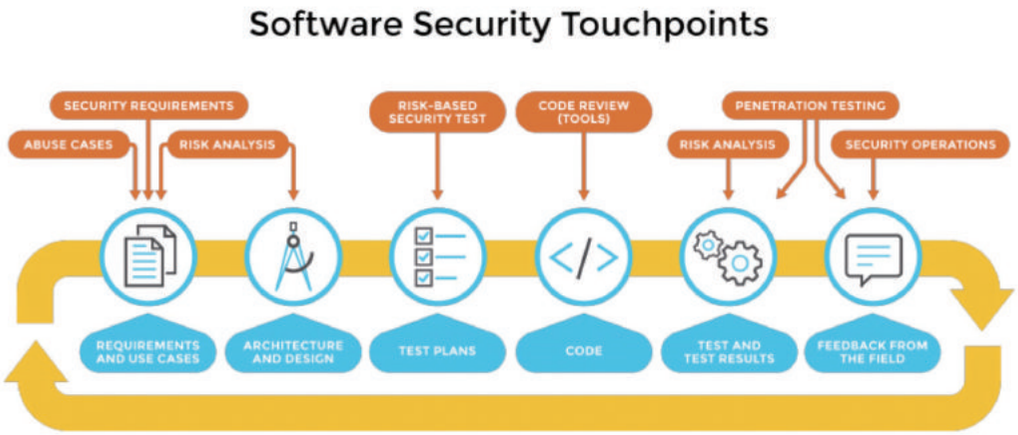
以前陣子的 ATM 盜領事件為例，組織內部有負責領錢的車手、負責洗錢的財務長、負責撰寫惡意木馬程式的程式研發人員、及確保木馬程式正常運作的品管測試。其實這樣的駭客惡意攻擊皆有一個共同公式：

攻擊（Attack）＝動機（Motive）＋方法（Method）＋弱點（Vulnerability）

動機我們雖然無法防範，但針對弱點我們卻可盡力避免——透過源碼工具檢測、人工滲透測試，以縱深防禦（deep in defense）有效降低惡意攻擊。

對岸網軍亦曾多次侵入國內諸多單位，一切的運作皆來自 1983 年的 ARPA（Advanced Research Projects Agency Network）研發出來的 TCP/IP 架構，當初設計未思考到惡意運用，對 OSI 7 層架構設計上未包含資安保護機制，因此專業駭客便利用此特點竊取資料，將竊取的封包切割傳輸，確保低流量運作以躲避特徵辨識防護設備偵測阻擋，直到蒐集完成後再重新組裝還原封包進一步竊取機密資訊。

另有一例則是在 2013 年 4 月 23 日下午由 Twitter 敘利亞電子軍（Syrian Electronic



↑ Cigital 機構提出的 SSDLC 軟體安全生命開發週期架構 。

Army,SEA）放出的假消息[1]：「白宮發生兩次爆炸，歐巴馬受傷！」，這則消息造成美國股市暴跌，短短幾分鐘內市值蒸發幾億美元，快速反應的股市交易來自交易系統蒐集分析相關新聞資料，再透過語意分析、人工智慧決策演算法和 FinTech 來進行買賣出交易的媒合，因此造成大量賣出，難以想像的因為一則假消息，造成可能比戰爭更可怕的經濟損失。

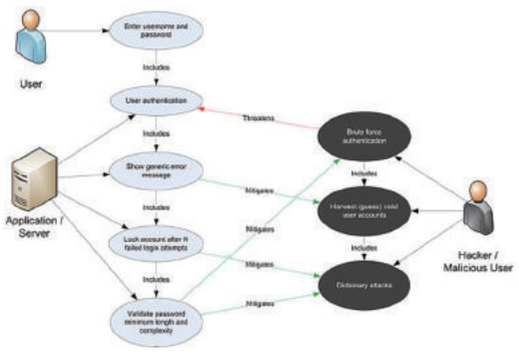
由 SSDLC 著手的資安觀念[2]

孫子兵法云：「知彼知己，百戰不殆；不知彼知己，一勝一負，不知彼不知己，每戰必殆。」通常攻擊者會竭盡所能找出系統所有弱點，如何減少弱點（Vulnerability）並加入安全（Security）元素也將成為這場戰役關鍵要素。

從 SSDLC 說明應用程式開發時該注意的安全事項如下：

需求面

探討相關資安需求，對外介接服務需透過加密傳輸，遵循單一登入準則（SSO），同時加入一些惡意的使用情境（Abuse Cases）。



↑由駭客角度使用系統分析可能造成的問題環節 。

架構和設計面

現今網頁開發最新技術架構為 MVC（Model、View、Control），對外服務採用 Web API，技術架構需考量的有：

- 威脅來源
- 目前使用的 Framework 是否有被通報資安問題？

[1] 參考資料： AP Twitter account hacked, makes false claim of explosions at White House (http://www.theverge.com/2013/4/23/4257392/ap-twitter-hacked-claims-explosions-white-house-president-injured)
[2] 參考資料： Building Security Into the SDLC Without Impacting Velocity (https://www.cigital.com/blog/building-security-into-the-sdlc/)

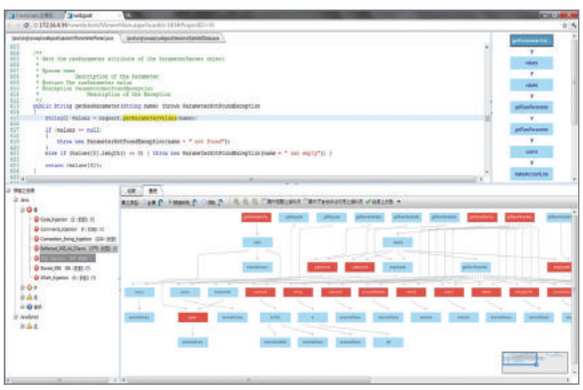
- 誰來使用 Web API？使用時要先登記註冊並加以記錄稽核軌跡，且規範敏感性關鍵字不可寫在 Client Side 以防外洩，如：javascript 註記有關密碼帳號、jQuery 呼叫 Web API 位置。

測試計畫

每個功能的單元測試需依惡意情境傳入測試資料並檢視是否影響系統，如 SQL Injection、Command Injection、Cross Site Scripting 測試腳本，測試計畫也需包含災難復原計畫。

撰寫程式

透過源碼檢測工具進行 Code Review，藉由工具舉報弱點程式碼，可讓開發人員於開發過程中即早處理資安的問題程式碼。



↑源碼檢測工具 。

測試及測試結果

依據測試計畫執行開發完成的功能，其中含單元測試及整合測試或滲透測試，並可透過黑箱檢測工具模擬駭客攻擊手法，將測試的結果提報。滲透測試則需長時間及專業資安人員執行，通常會在系統功能完備後才會進行，一方面考量測試成本，再者也考量測試時間。

↑黑箱檢測工具 。

回饋相關資安弱點問題

開發人員依據測試人員的測試結果進行修正，此時 SSDLC 則一直不斷循環，弱點數量也會跟著減少，目前也可透過 CI（Continue Integration）工具整合自動化作業減少人工負擔，並藉扎實導入 SSDLC 有效減少系統弱點。

不過再安全的系統也無法避免因人為疏失造成的資安問題，以全球頂尖駭客凱文・米特尼特為例[3]，其慣用手法為先對實行詐騙的公司進行資料偵查，含系統代號、部門名稱及主管名稱及分機號碼等，以利在社交過程中取得對方信任並降低防備，並利用人性弱點去騙取相關機敏資訊。

資安大師布魯斯・施奈爾曾說：「資訊安全是一個過程，而不是產品。」這次我們從生活上面臨的資安事件談到由軟體安全生命開發週期著手的資安概念，一方面除了是想提醒軟體開發從業人員，也希望可讓使用者了解如何自我保護以及防範管理。下期論壇則將從 OWASP（Open Web Application Security Project）Top10 著手，談安全程式開發的教戰手冊，希望大家可以有所收穫。

[3] 參考資料：悅知文化出版【駭客人生】Kevin Mitnick 著，鐘協良譯