應用程式安全的作業方法：10 個步驟保護敏捷開發

文章來源：Checkmarx blog 翻譯整理：叡揚資訊 資訊安全事業處

在敏捷快速發佈的開發環境裡，安全審查與測試就像個阻礙成功的要素。該如何滿足敏捷的快速要求且不捨棄安全的作法呢？以下整理十個做法有助於實現安全的軟體開發生命週期（SDLC）的敏捷環境。以下做法的理念皆盡可能以簡化並減少不需要的負擔。

1. 成為流程的一部份

安全需求應考慮視為開發中的檢核點。每個基準點皆需完成才能進到下一階段的敏捷流程。每階段皆訂有安全的里程碑需要達成。發佈計劃中需明訂安全設計等級。包括以下幾個方面：

- 開發安全程式碼：如使用 API
- 安全技術：定義要使用的技術，如 Maven process 或系統中的安全測試
- 安全功能：法令法規要求，如追蹤 Cookies 以符合英國隱私法規

2. 在每個產品中使用安全的 API

分為二個階段：

1. 使用安全的工具如 OWASP's Enterprise Security API (ESAPI)

ESAPI 是一個可以讓開發人員輕鬆使用的工具包。它提供多種實用工具如驗證工具、加解密工具及隨機工具。透過使用 ESAPI 開發人員不需要再花額外的時間研究安全功能即可實作。

以雜湊 (Hash) 來說，不再需要開發人員額外添加鹽 (Salt)，鹽已成為 ESAPI 設定的一部份。開發人員可以輕鬆的利用 API 來達成。

特別是驗證工具，主要可用來預防常見的 Web 應用程式弱點，如資料隱碼攻擊 (SQL Injection) 及跨站腳本攻擊 (XSS)。每個組織都需要去評估這些驗證方式。有些可能會利用 Web Server 中的控制項，如 Tomcat filter。其他可能會傾向於對每一個輸入做驗證。這兩種都有單位採用，但我們發現其實較多的單位會選擇在每一個輸入時做驗證，這可歸納於二個原因：

- 使用正規表示式用來驗證輸入其設計可以盡可能單純化，以避免效能上的問題。
- 當不幸真發生問題時，也只需調整輸入驗證。另一方面若使用 Web Server 中的控制項需要測試整個系統以確保作業正常。

在我們的研究中採用更進一步的方法。在特定的驗證中不回傳 True/False 的布林值，但如果輸入為無效時回傳 Null。在這種情況下可以避免開發人員在程式中使用錯誤的值。

2. 確認開發人員使用的是正確的 API

確保開發人員對於每個輸入，都有使用由安全團隊提供的正確驗證器。這需要進行安全性測試以確認開發人員未採用不當的 API。這可透過靜態原始碼測試 (SCA) 來達到。

3. 在程式碼管理流程中整合靜態原始碼測試

要簽入的程式碼未通過安全政策，則開發人員無法繼續建置。為了跟上節奏快速的開發環境，開發人員必需在短時間內消化這類安全政策。

面對這樣的挑戰需在不同的開發階段整合 SCA。特別要注意的是：

- **當自動建置工具 (如 Maven) 與 SCA 整合時：**
  企業組織通常以兩種模式執行 SCA
  1. 第一種：每種有異動時即進行差異的 SCA 分析。在這情況下只有在上次檢測至今有修改的部份會進行檢測。
  2. 第二種：利用夜間的時間對整個專案檢測完整的安全掃描。若建置失敗，開發人員需先修復安全缺陷才能繼續開發。
- **建置管理與持續整合伺服器 (CI) 呈現 SCA 的結果：**
  當有 SCA 警示時，對於開發人員可更有效的發現 / 識別弱點。
- **知識庫的整理，讓開發人員了解應如何修復弱點：**
  因此 SCA 工具若能包含一個知識庫並描述弱點及適當的修補建議有助於安全政策的推動。

4. 清除任何高、中的弱點

不要發佈任何包含高或中弱點的產品。這些缺陷應在建置過程中處理完。也就是說如果開發人員發生高安全弱點應停止建置，直到弱點被修復後才能將其封裝為產品。

5. 使用自動化動態安全測試

在產品中進行動態測試可分為 positive / negative 兩種：

- **使用 positive 測試輸入驗證：**舉例來說，一個 positive 測試會去驗證表中單輸入電子郵件的欄位是否包含 "@" 及 "." 的符號，同時沒有其他特殊字元。
- **negative 測試正是補足 positive 測試：**延續上述的案例，電子郵件的欄位中若被嵌入 SQL Injection 將會在 negative 測試中發現。因此這兩種在動態測試中是互補的。

6. 執行滲透測試

聘請專家及客戶進行擔任滲透測試人員：

- 由外部廠商進行最終產品的滲透測試，包括自動及手動測試。
- 允許客戶執行滲透測試，並使用社群互動達到效益。組織仍需要遵循必要的步驟達到發佈安全的產品。許多客戶也樂意參與第三方產品的滲透測試。對本身的好處即可得到客戶的信任。談到軟體服務 (SaaS) 產品，最關鍵也最重要的即是贏得客戶及供應商的信任。

7. 聘請專案的安全技術團隊並展現他們的價值

即使已有具有規模的安全團隊，但很明顯的開發團隊遠多於安全團隊。為了增加安全的推廣將技術領袖安排為 security champions。與 R&D 合作在每一個 Scrum 會議中皆包含安全的部份。

8. 定期對開發人員訓練

這裡的重點不是一定要建立一個正式的訓練過程，而是開發人員需了解 Web 應用程序安全課程。還有一些其他的訓練方式：

- **利用 SCA 的弱點知識庫提供開發人員安全的知識：**透過幫助開發人員了解風險及其解決方案，增加安全的意識，讓開發人員從不同的角度審查安全及程式碼。
- **資安政策應對開發人員開放：**一旦安全成為必定的程序，從開發到 Q&A 將累積不少安全的議題。資安團隊應提供相對應的政策以解決開發人員的顧慮。

9. 提供討論安全的協作平台

即使已有具有規模的安全團隊，但很明顯的開發團隊遠多於安全團隊。為了增加安全的推廣將技術領袖安排為 security champions。與 R&D 合作在每一個 Scrum 會議中皆包含安全的部份。

10. 從小做起，但要將格局擴大

許多做法在實務上需要管理階層及高層的支持，特別是當發現高或中的弱點要停止建置時。我們意識到這並不是件容易的事。根據許多公司的經驗，以下幾個步驟相當有幫助：

- **先挑一個小專案並把它轉為成功案例：**聽聽 R&D 在這個過程中從錯誤中學習並改善的過程。
- **從一個成功案例再轉移到另一個新的專案：**持續改善並從錯誤中學習，再持續建立 2~3 個成功案例。
- **審查由客戶提出的安全弱點：**比較成功案例與其它專案的弱點數量及做法。 讓管理階層了解這些弱點是如何干涉正常產品的交付及維護。
- **進展至既有大型專案：**一開始不要因發現安全弱點而停止建置。這階段可以先確認差距有多少，再來建立如何修復的程序。
- **只有在有把握後，才進行既有大型專案的弱點修復：**只有了解系統中的弱點，並正確選擇適合的安全 API(如 ESAPI)，同時需驗證不會影響到產品本身，即能一步步完成。
- **繼續建置中的大專案：**這是最終的目標，此階段安全應該已整合為敏捷過程中。在該步驟中， 驗證器應該已經被封裝，並設置於一個框架內。

更多 " 應用程式安全的作業方法：10 個步驟保護敏捷開發 " 部落格文章，請參考：
http://www.checkmarx.com/white_papers/the-appsec-how-to-10-steps-to-secure-agile-development/