應用系統安全與品質必備工具 持續整合平台 CASIP (原 CI Server)

CASIP (Continuous Application Security Improvement Platform)

導入 CI Server，藉其整合能力，用系統設定來取代客製化的程式碼撰寫

撰文｜叡揚資訊安全事業處 資深專業顧問 陳惠群 博士

我們曾經發現客戶前後兩次的 Fortify SCA 原始碼檢測報告，差異頗大；追查後發現，原來是第二次送檢的原始碼內容有錯誤或是遺漏。倘若送檢的原始碼都是自動簽出、並且通過建置測試，才進行原始碼檢測，發生前述狀況的機率必然大幅降低。

應用系統上線前後，才發現品質或安全問題，將意味著必須投入極大量的人力與成本來進行改善或修復。因此在開發初期，就應當將品質與安全，納入規格當中；並且確認系統設計、開發、測試、維運等階段，都符合品質與安全規格要求。

為使應用系統於開發階段，就開始檢查程式品質與安全，許多團隊會先以人工方式收集原始碼，再利用工具進行檢核 (如 Fortify SCA)。但這代表著須有額外的人力投入，以及人為疏失的風險。CI server 就是將以上人工作業，變成自動化機制，將開發人員每日開發的原始碼整合，利用下班時間，自動與程式品質或安全檢核工具串接，進行自動檢核，達到節省人力及避免人為疏失的效果。

持續整合 (Continuous Integration；CI) 是軟體開發最佳實務之一：通常在軟體專案過程中，系統整合與測試都是留在後半段，萬一到了這個階段才發現規格不符，修改成本必然十分驚人。持續整合試圖解決這類問題，在專案初期，就開始進行介面整合與相關測試，並且持續進行。這種重覆不斷的程序，很明顯地需要搭配適當的工具，才能落實。

CI 的初衷是確保程式間合作順利，最基本的驗證方式自然是通過開發工具編譯 (Compile) 與繫結 (Link) 的靜態檢測，確認程式間介面呼叫及參數型態符合規格；更進一步則可藉由撰寫測試程式，例如單元測試 (Unit Tests)，來確認執行時期的合作狀況。因此持續整合平台 (Continuous Integration Server，簡稱 CI Server) 應具備下列四大特色：

特色 1　與多種版本控管軟體整合　自動取出原始碼

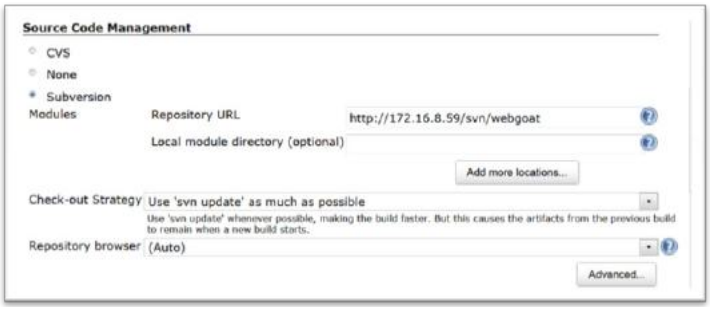

圖 1：設定由 Subversion 取回專案原始碼

無論是廠商或是業主，只要有軟體原始碼管理需求，多半都已導入版本控管 (Revision Control) 軟體，像是 TFS、PVCS、CVS、Subversion、Git 等等。CI Server 最基本的功能，便是由這些版本控管軟體取回原始碼，存放至 CI Server 預設的工作區域 (Workspaces)，如此才能順利進行後續步驟。通常 CI Server 預設取回最新版本的原始碼，當然也可以依據版本號碼或是標籤 (Tag)，指定取回特定版本。

持續整合平台 CI Server 對應用系統安全與品質之重要性

應用系統上線前後，才發現品質或安全問題，將意味著必須投入極大量的人力與成本來進行改善或修復。因此在開發初期，就應當將品質與安全，納入規格當中；並且確認系統設計、開發、測試、維運等階段，都符合品質與安全規格要求。

為使應用系統於開發階段，就開始檢查程式品質與安全，許多團隊會先以人工方式收集原始碼，再利用工具進行檢核 (如 Fortify SCA)。但這代表著須有額外的人力投入，以及人為疏失的風險。CI server 就是將以上人工作業，變成自動化機制，將開發人員每日開發的原始碼整合，利用下班時間，自動與程式品質或安全檢核工具串接，進行自動檢核，達到節省人力及避免人為疏失的效果。

特色 2　整合建置工具　自動建置執行碼

當 CI Server 簽出原始碼後，最基本的檢測就是對程式碼進行編譯，來檢查每一支程式的結構及語法使用；或是進行所有程式的建置 (Build)，來檢查整個專案的完整性。

CI Server 並不懂得如何編譯 C# 或是 Java 程式，而是具備與程式語言開發工具整合的能力，能夠在簽出原始碼後，呼叫正確的建置工具 (Build Tool)，例如 MSBuild、Ant、Maven，或是編譯工具，例如 gcc 或是 javac，並且給予適當的參數值，例如原始碼存放路徑。

每日建置 (Daily Build) 也是軟體開發最佳實務之一。對於軟體開發團隊而言，每天下班後，將當天工作成果集中起來，然後進行自動建置，不僅能夠及早發現問題，當第二天早上，打開電腦，收到一封專案建置成功的通知信時，也有激勵士氣的效果。

圖 2：設定使用 Ant 來建置專案

特色 3　整合檢測工具，自動檢測原始碼或執行碼

CI 平台提供簡單但功能強大的整合能力：只要具備 CLI(command-line Interface；命令列控制介面) 的軟體，CI 大概都能與其進行整合。以 Fortify SCA 為例，便提供了 CLI，使用者可以在系統命令列執行 sourceanalyzer 這支程式，來執行原始碼弱點掃描。當然 Fortify SCA 檢測工具，通常不會與 CI 平台安裝在同一台伺服主機，因此必須配合遠程呼叫來啟動檢測。

圖 3：透過命令列介面來執行 Fortify 檢測

特色 4　整合系統工具，進行自動佈署（上版）

應用系統上線佈署，有一定的作業程序，例如檢查佈署內容的完整性 (Integrity Check)，確認使用合適的帳號、停止執行中的服務、將佈署內容移至正確的位置、重啟服務等。這些動作，多半可以透過系統命令完成，並且可以進一步透過撰寫佈署腳本 (script)，將這些動作串連起來。

當 CI 平台完成自動建置後，其工作區會同時存在原始碼和目的碼 (執行碼)，只要經過適當的安排，CI 平台可以輕易備妥佈署內容，接著呼叫佈署腳本，執行佈署動作。

其實早在 2006 年開始叡揚著手導入 CMMI，期望提升軟體專案品質時，引進了一系列的工具軟體；其中有一套 CruiseControl.NET，正是 CI Server，能夠由版本控管系統自動簽出專案原始碼，並且結合許多

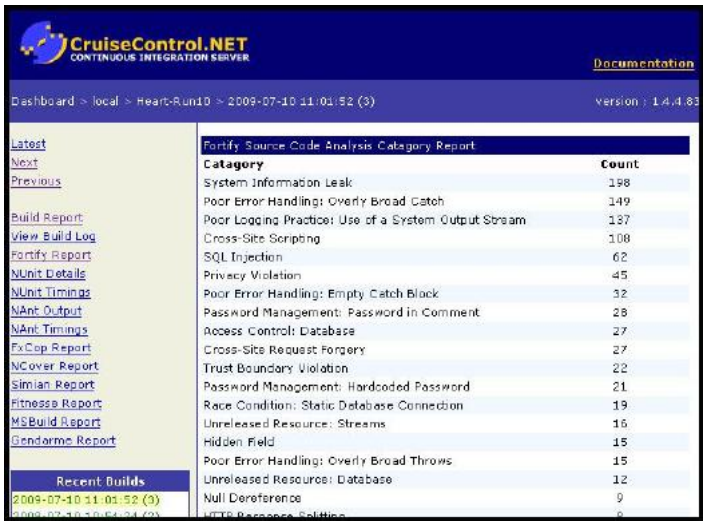

圖 4：CruiseControl.NET CI Server

品質檢測軟體，自動進行檢測。當時正好叡揚開始代理 Fortify 產品，因此也試著將原始碼安全檢測納入，測試 CruiseControl.NET 的整合能力，如上頁圖 4 所示。

叡揚累積多年內部使用 CI Server 的實際運作經驗，協助專案開發團隊，管控專案進度與品質。回想起這段內部推動過程，引發我們思考，如何將這個成功經驗，移植給客戶。例如我們曾經發現客戶前後兩次的 Fortify SCA 原始碼檢測報告，差異頗大；追查後發現，原來是第二次送檢的原始碼內容有錯誤或是遺漏。倘若送檢的原始碼都是自動簽出、並且通過建置測試，才進行原始碼檢測，發生前述狀況的機率必然大幅降低。

也有不少客戶，特別是已經導入服務管理或是資安管理的單位，則有將原始碼檢測與管理流程整合之需求。事實上，叡揚曾經配合客戶的表單系統，開發專屬程式，進行原始碼自動檢測。然而這類客製化服務，往往不易擴充與維護；因此我們也建議客戶可以導入 CI Server，藉由其整合能力，用系統設定來取代客製化的程式碼撰寫。

圖 5：Jenkins CI Server