Gartner 在 " 靜態分析產品 " 象限，Checkmarx 是唯一一家榮獲最高分的源碼檢測產品供應商
— 2014 ~ 2019 年度 SAST

Checkmarx

01

Checkmarx 使應用系統安全更容易

Checkmarx 是一款高準確率（低誤報）且靈活的源碼檢測軟體，可識別主流開發語言的數百種安全漏洞及品質缺陷，該軟體可以獨立運行，也可以有效整合到SDLC軟體開發生命週期過程中，去加速軟體安全漏洞及品質缺陷的發現和修復過程。該軟體可以獨立運作在企業內部，也可以利用私有雲或公有雲方式使用。

Checkmarx 作為軟體安全弱點檢測解決方案供應商，已經在業界確立了其領導地位，其 Static Application Security Testing（SAST）源碼靜態安全測試方案為眾人所知。客戶囊括了世界 10 大軟體供應商中的 4 家，數百家財富 500 大公司以及各個行業的諸多中小企業。

軟體開發生命週期

Checkmarx 可以協助企業整合源碼靜態安全測試 Static Application Security Testing（SAST）到軟體開發生命週期 SDLC（Software Development Life Cycle）中。我們整合了最常見的建置管理工具、問題追蹤工具、以及 IDEs 整合開發環境。

內容

此平台由集中的控制介面所組成，用以設定與監控應用系統安全測試，並提供儀表板顯示測試結果與風險指標，讓整個軟體開發生命週期都可以管理注意安全策略。

AppSec 開發培訓課程（CxCodeBashing）

Foreword 前言

Codebashing 開發培訓課程主要希望養成開發人員對於軟體安全的觀念與重視，讓開發人員能將安全掌握在自身手中並樂於執行。利用即時的弱點挑戰讓學員可設身處地的了解原因以及快速了解如何撰寫安全的程式碼。

根據美國國土安全局發布的論文 (註 1) 指出，軟體安全基本上是軟體功能的問題之一，必須在軟體開發生命周期中有系統的管理及學習。

人們普遍認為安全來自於軟體工程師，但依據 Node.js® 和 Sqreen 的研究 (註 2) ，六成的工程師對於自己開發應用系統的安全強度沒有信心。這與 SANS 於 2016 年的應用程式安全報告 (註 3) 的結果相符，該報告指出最核心的挑戰在於缺乏應用程式安全（AppSec）的技能、工具與方法。

而為什麼會缺少 AppSec 的技能？

從 CloudPassage (註 4) 的研究可以發現在美國 computer science 學程中只有一間要求安全概念課程才能畢業。依據 StackOverflow 的統計 69.1% 的工程師其實是自學而來的。這告訴我們若企業希望自己的工程師交付安全的程式碼，需要提供他們一流的安全程式撰寫教育訓練（Secure Coding Education, SCE）。

進一步來說，若企業也認同，依據 SANS 的報告企業認為開發人員的訓練中比起弱點的掃描 / 檢測，更加需要加強的是應用程式安全（AppSec）的技能。不過現實生活中即使企業安排了各式的安全教育訓練與方法，技能的落差仍有漫長的一段路需要努力。

本指南希望能降低企業所需之安全程式與開發人員技能間的落差。將引導您了解需要知道的所有事項，以確保軟體工程師獲得最有效的 SCE 並符合實際需求。

註 1 https://www.us-cert.gov/sites/default/files/publications/infosheet_SoftwareAssurance.pdf
註 2 https://www.businesswire.com/news/home/20171109005210/en/NodeSource-Sqreen-Survey-Quarter-Node.js-Developers-Form
註 3 https://www.sans.org/reading-room/whitepapers/analyst/2016-state-application-security-skills-configurations-components-36917
註 4 https://www.cloudpassage.com/company/press-releases/cloudpassage-study-finds-u-s-universities-failing-cybersecurity-education/
註 5 https://insights.stackoverflow.com/survey/2016#developer-profile-education

Understanding the Developer's Perspective 了解開發人員的想法

現今快速的開發環境中，需要快速交付並持續整合、持續交付（CI/CD），開發人員最寶貴的資源就是時間。撰寫安全的程式碼會降低開發人員的速度，這所帶來的問題可能會導致忽視安全的議題。正因如何企業必須考量與 DevOps 環境可以匹配的源碼檢測 Source Code Analysis （SCA）方案。

在解決開發人員的安全技能落差時要注意的另一個重要因素在於開發人員的主要工作為撰寫「程式碼」；很少在應徵工作時條件為「安全程式撰寫 / 交付」。安全程式撰寫就慢慢演變為「如果有時間再處理（Nice to have）」，但往往時間都是不夠的；而軟體工程師在評估績效時通常是透過速度與解決的 Bug 數，而不是解決安全漏洞的數量。

綜合上述所言雖然對開發人員充滿同情，但仍需要求開發人員提交安全的程式代碼。為了實現這點，需要改變的第一件事即是開發團隊的領導者需要在開發的過程中處理 / 修正安全弱點。而開發人員也需了解目標為交付無 Bug 且預先考量妥善防範的程式碼。此時就需要落實安全程式撰寫教育訓練以達到這個目標。

Deciphering Developer Secure Coding Education 分析開發人員安全程式撰寫教育訓練

教育訓練常見的方式為影片、定期課程與線上課程。這些活動多半被列為待辦清單或例行公事，並非將其認定為安全程式開發的工具。因此多數此類活動在進行時，開發人員的思緒都在於其他工作的執行。也因如此參與的學員也會以較不重視的態度參與。有鑑於此，如何讓安全程式撰寫教育訓練更容易推行？可能要從課程的模式改變，或許改採用遊戲化、角色伴演的方式進行，將會是更能夠引發興趣及提升學習效率的一種方式。

Gamification 遊戲化訓練

遊戲化並非指的是玩遊戲，而是仿造遊戲的設計原則與元素。雖然遊戲化在教育訓練中並非是新的議題，但多數的安全程式撰寫教育訓練並未落實。當學員處在享受的環境中學習，其成效會更好。開發人員長時間都在撰寫程式碼，依據參加過課程的數千位工程師的回饋，透過程式碼的閱讀、修改會讓開發人員更能接受這樣的課程。若需在企業中落實安全程式撰寫教育訓練，請注意以下四點，以確保學員都能有效的參與學習課程。

Make it Interactive 一定要互動式的

Chief Learning Officer 提到：「點擊的頻率並不能代表學習內容有吸引力，有可能只是學員想快點完成」。這種情況在組織強制性的課程中非常常見，學員並不理解課程的內容，只是想提早完成，儘早回去完成繁重的工作，而不是接受重要的內容，這將導致無效的課程與測驗。

為什麼需要互動式學習，課程中包含的故事與實例是吸引學員參與的重要部份。

故事中創造了一種情境，讓學員切身參與，可提高課程吸收力與印象。此外，若課程僅需單純的點擊式互動，不易增加課程的互動性。　為了提高互動性，學員需更加關注內容，提供實務學習的機會；實際執行操作相較聆聽或閱讀的效果更好。

Tell a Story 勾勒情境

故事背景介紹、角色扮演可以刺激學員的反應，當現實生活中若面臨相同情況，將需如何應對？安全程式撰寫教育訓練若以條列式的問題、答案易讓學員產生例行公式的厭煩感。而情況、人物、所面臨的問題可避免這個問題。依據故事情節（攻防實例、需解決的漏洞），有助於記住所學的內容。而故事也是許多遊戲的精髓，往往是一種有趣的催化劑，與遊戲化是切不可分的。

Keep it Short 精簡

近年來因 3C 產品的發展，人們的能夠專注在一件事上的時間越來越短 (註 6)。提供的資訊保持精簡是最好的方式。如同演講中的使用的 PowerPoint 一樣，提供簡短重要的內容，以減少過多不必要的資訊。而這正是我們課程的原則，以最有限的時間、資源，提供最重要的安全資訊與防範作法。

Ensure They Win 確保參與者贏

依據 Dr. Ian Robertson 著名的研究「The Winner Effect」(註 7)。最容易被低評的是大腦潛力。當滿足學員參與的信心，大腦會釋放些刺激因子增加腦中的多巴胺活性，讓學員可以擁有信心，積極主動的學習。

Contextual Learning 情境體驗

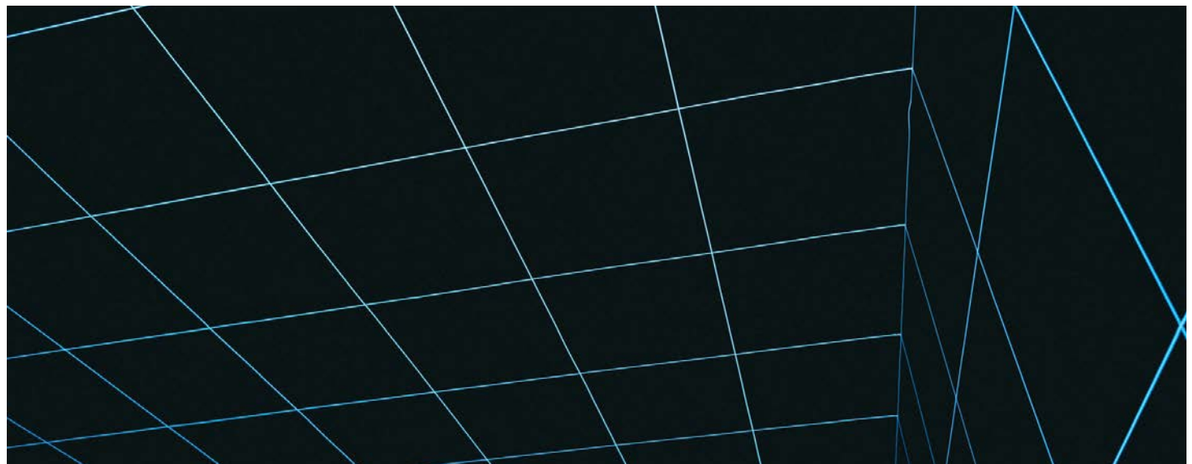
準備精彩、遊戲化的教學對於開發人員實際使用與學習是最重要的。而課程的學習與投入工作的時間往往是衝突的，盡量以最短的時間提供開發人員對於弱點修改的印象，當實務上發生時即可想起對應的方法。我們發現，開發人員在編寫代碼時可以持續訪問我們的培訓課程，鼓勵他們在遇到安全漏洞時回來查詢弱點的資訊。最後需持續更新程式語言版本。安全程式撰寫在各語言間有些許差異，但持續更新是所有語言都需注意的。

In conclusion... 結論

我們得到的結論是，以短期、互動、有故事性對於教育訓練來說是重要的元素。不要忽視「贏」對於課程的重要性，用勝利作為課程的結束，如成功解決弱點。確認開發人員的培訓是有效的；同時也因讓人員感覺良好，會自願的去找尋解決方案，以探索新的弱點挑戰。

Summing Up 總結

希望您在採購安全程式撰寫教育訓練前，清楚的了解課程的行式以及對於開發人員的幫助，本課程是藉由 Checkmarx 研究團隊豐富的經驗，並以遊戲的方式進行讓課程不再是枯燥的例行公事。若想了解更新請嘗試 CodeBashing 的線上課程，學習對於開發人員的撰寫實務。經歷課程後仍對處理方式有印象且實作它們。



支援語言

Java

(Advanced) Java

Front End

(Java & .NET) Backend

HTTP

.NET

PHP

Node.JS

(on Rails) Ruby

Go

C/C++

Android

iOS

Python

Scala

效益總結

Checkmarx 軟體安全平台是功能強大的軟體安全測試工具，提供以下重要不可或缺的好處。

企業級的軟體管理平台

集中設定與管理

單一的風險儀表板

量身定作，優化調整規模

GSS IT & Security

安全弱點管理救星

您知道您所使用的資安檢測工具該如何正確的使用嗎？

依據叡揚資訊多年協助客戶導入資安工具的經驗，多數的客戶在面臨資安工具時，常有以下幾個情境：

- 不了解工具如何正確操作
- 收到很多的檢測報告，但對於修復不知所措

安全性弱點修復的挑戰

對於程式開發人員來說，安全性弱點總是又多又綁手綁腳，在不理解其原理的情況下，難以說服這些是需要花時間成本進行修復；叡揚資訊提供客戶服務平台，將針對詢問的弱點，說明其原理以及可能的危害，藉此讓相關的處理人員了解所花費的時間成本是必要的，同時也提供建議的處理辦法，以利後續降低弱點的作業。

安全性弱點管理的挑戰

對於不同系統的檢測結果，或許可能有些問題的相似的，而這之間的修改 Know–How 是值得共享的，此時可藉此平台的分享歷史諮詢記錄來達到知識的傳承；利用搜尋功能來快速找尋解決辦法，提供修改的效率。

諮詢記錄分享