Checkmarx

源碼安全檢測

在駭客不斷增加的風險中保護世界應用程式安全的工作而被公認為應用程式安全領導者

2019–2022 年榮獲 GARTNER® PEER INSIGHTS™ 應用程式安全測試客戶最佳方案

2018–2022 年被 GARTNER® 應用程序安全測試魔力像限™ 評為領導

IT中台同行獎–應用安全測試

CDM 全球信息安全獎——應用安全市場領導者

網絡安全全球卓越獎 – 應用程式安全和測試

網絡安全卓越獎 – 最佳網絡安全公司

Checkmarx 使應用系統安全更容易軟體開發生命週期

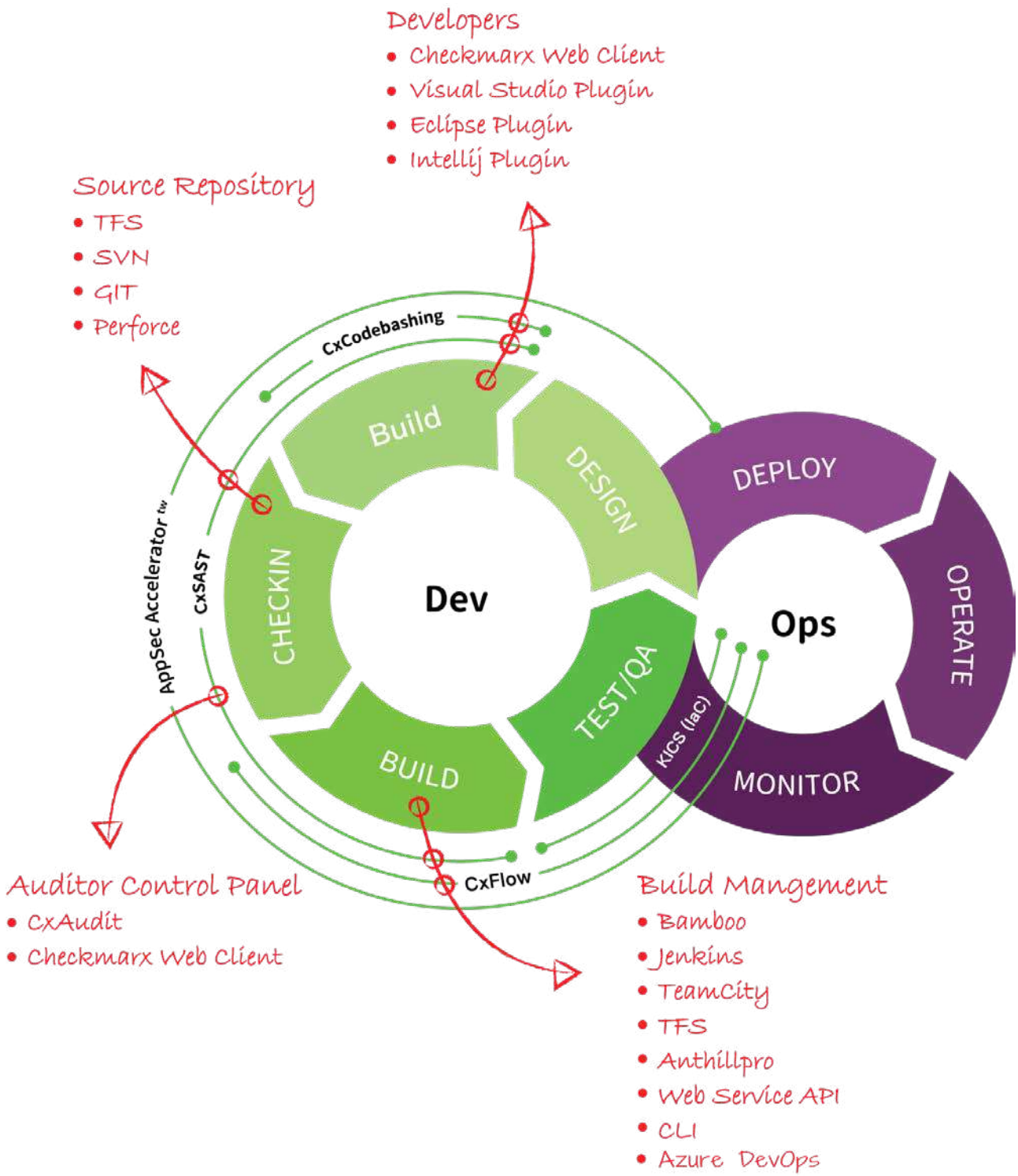
Checkmarx 作為軟體安全弱點檢測解決方案供應商，已經在業界確立了其領導地位，其 Static Application Security Testing（SAST）源碼靜態安全測試方案為眾人所知。客戶遍及全球，超過 1400 多個大型企業都是使用 Checkmarx 來確保自家的軟體安全。

軟體開發生命週期

Checkmarx 可以協助企業將源碼靜態安全測試 Static Application Security Testing（SAST）整合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。



源碼靜態安全檢測（CxSAST）

CxSAST 是一種靈活、準確的企業級靜態分析解決方案，能夠識別自行開發程式碼中的數百種安全漏洞和弱點；支援 20 多種程式碼、框架與腳本語言。

安全軟體開發生命週期

Checkmarx 能夠幫助組織整合源碼靜態安全測試 Static Application Security Testing（SAST）到軟體開發生命週期。比如，整合到最常用的版本控管工具、建置管理工具、問題追蹤工具、以及 IDEs 整合開發環境。如果 Checkmarx 無法立即整合到軟體開發生命週期的某一組件，只需透過 API 便可輕鬆解決該問題。全面的 SAST 模型優勢在於：

- 資安團隊可以專注於資安政策，使用 Checkmarx 達到自動化執行資安政策。
- 對近期新增程式片段進行快速安全測試，在開發人員記憶猶新時對所有新發現弱點進行修補。此舉不僅大幅地降低成本，還省去了在產品上線日逼近時需修復大量安全隱憂的煩惱。

支援業界標準

MISRA

CWE Mitre CWE

HIPAA

OWASP Top 10

BSIMM

SANS 25

PCI DSS

OWASP Mobile Top 10

FISMA

OWASP ASVS

CxSAST 能找出數百種弱點，包括常見弱點

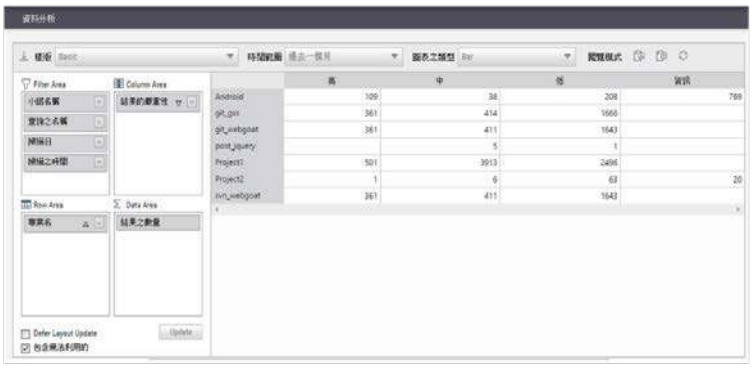
SQL Injection	Cross-site request forgery	Log forgery
Cross-site scripting	HTTP splitting	Unreleased resources
Code injection	Session Fixation	Unvalidated input
Buffer Overflow	Session poisoning	URL redirection attack
DoS	Unhandled exceptions	Dangerous Files Upload
Parameter tampering		Hardcoded password

CxSAST Web 介面

資安人員及開發人員必須審查已列舉出來的弱點並決定最佳的修復辦法。Checkmarx Web 介面為能提供最佳的用戶體驗，它能呈現攻擊流向及資料從輸入到執行的流程，點擊每個節點都能顯示出相關的方法和弱點處。

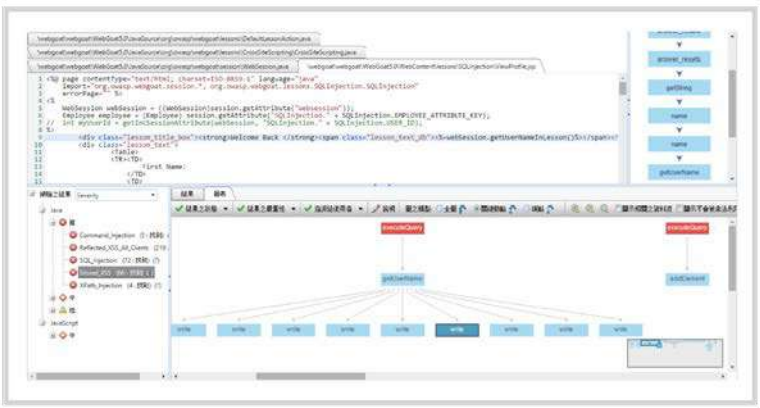
儀表板（Dashboard）與報表

使用 Checkmarx 可輕鬆分析資料並輸出報表。您可以使用預設的資料分析報表，或是透過樞紐分析的拖曳介面來設定所需的參數產生圖形化的資料，隨後可匯出 PDF 或 Excel 格式。



加速弱點修復工作

Checkmarx 不僅能識別弱點所在。除了表列的結果外，透過圖形演算法結合攻擊向量，直指多處攻擊流程的共同節點（最佳修復位置）。利用圖表（Graph View）讓開發人員可得知需修復最少的節點，即可達成全弱點修復。



Checkmarx 產品獨特之處

無需編譯、開發初期即可檢測

我們能夠檢測未經編譯的原始碼，意味著在開發周期的初期即能檢測，而此時恰是偵測安全漏洞的最佳時機，也意味著您不必擔心程式需經過編譯、完成編譯後才能檢測，只需於產品中放入程式片段即可。

源碼未變動則無須重複掃描

如果僅有數行程式有變動，通過 Checkmarx 獨一無二的差異掃描（incremental scan），就無需重複掃描整個專案。我們會分析自上次掃描後有變動的部分及其相依的文件，然後僅對這些進行掃描，如此便可快速得出結果，對於高速的敏捷開發環境尤為有用。

檢測規則透明且可客製化

Checkmarx 的產品公開查詢規則，意味著您可以清楚的看到 Checkmarx 的掃描內容與掃描方式，同時，您也可根據特定的環境快速做出修復，並添加自行訂定的過濾方法，從而將誤報率和漏報率減少至可忽略不計的水準。進階的使用者往往會添加自己的查詢規則，利用 Checkmarx 輔助達成最佳撰寫實務、合規性及更多其他功能。

整合至現有軟體開發流程

Checkmarx 能非常靈活地整合至現有的軟體開發生命周期中，因此，您可以決定所需的安全政策，並且以自動化的方式實施。我們支援常用的版本控管工具、建置管理工具、問題追蹤工具以及 IDE 整合開發環境，使您能加速安全測試並確保最高效率地完成任務。

加速漏洞修復

Checkmarx 能做的不只是偵測識別原始碼漏洞。透過應用程式的整體資料流程，能偵測出漏洞關聯所在，利用「最佳修復點」您可一次修復大量漏洞，實現軟體修復最佳化。

涵蓋主流的程式語言

Checkmarx 設計架構可以容易、快速的支援新的程式言及構架。目前支援超過 20 個程式語言、腳本語言及通用框架（Framework），每年大約新增 2~3 個種語言。

成功客戶

ATLASSIAN

沒有比 Checkmarx 更簡單易用的工具了。重要的是，你無需整合到建置管理工具，只需要把程式碼丟給它就可以了。在技術支援方面，我們團隊極為滿意。儘管時差不同，技術支援服務仍非常專業、及時。

台灣某政府單位

支援語言較多，能培養工程師定期使用源碼檢測工具的習慣，比起專案尾聲再執行檢測工具更有效率，同時避免相同的弱點產生，讓工程師開發時更嚴謹，也能節省事後修復程式問題的時間與人力。

salesforce

salesforce.com 網站選擇 Checkmarx 的靜態源碼分析工具作為官方 Force.com 的軟體安全檢測工具，至今檢測超過 13 億行 source code。Checkmarx 確保了 AppExchange 所有的應用程式安全性都達到最高標準。

中央研究院 ACADEMIA SINICA

Checkmarx 涵蓋率及準確率較高、支援行動裝置 APP 且不需受限於開發環境版本的特性，最重要的是簡易上手、不用高複雜度的建置環境，使用單位學習快速。

支援的語言

Java, JSP, JavaScript, VBScript, PL/SQL, HTML5

C#, ASP, VB, .NET, .Net Core, VB6

Ruby, TypeScript, Perl, iOS, Android, Windows phone

Groovy, Scala, GO, Python, C/C++, PHP

Cobol (New), RPG (New), Dart (New)

輕鬆實現 DevSecOps 自動化（CxFlow）

Checkmarx 提供 CxFlow，讓開發人員能輕鬆整合 Webhook，透過 SCM（Source Code Management）觸發送掃事件，並將掃描結果回傳 SCM，降低開發人員工作壓力，達到同一平台完成送掃與問題追蹤作業。

過去的日子，開發人員已習慣於 SCM 的程式碼整合與事件（Event）管理方式，透過 CxFlow，開發人員在熟悉的平台上，便可透過 Webhook Event 進行送掃，無需變更現有作業流程，亦可在原來的平台上進行 Issue 追蹤，大幅的降低了開發人員在軟體適應度上的門檻，強化掃描與配合意願。

除了市面上常見的 SCM Event 之外，CxFlow 亦可作為 CLI（Command-line interface）工具嵌入 CI/CD Pipelines 內，完善 SSDLC 流程，輕鬆駕馭 DevSecOps。

三個一定要試 CxFlow 的理由

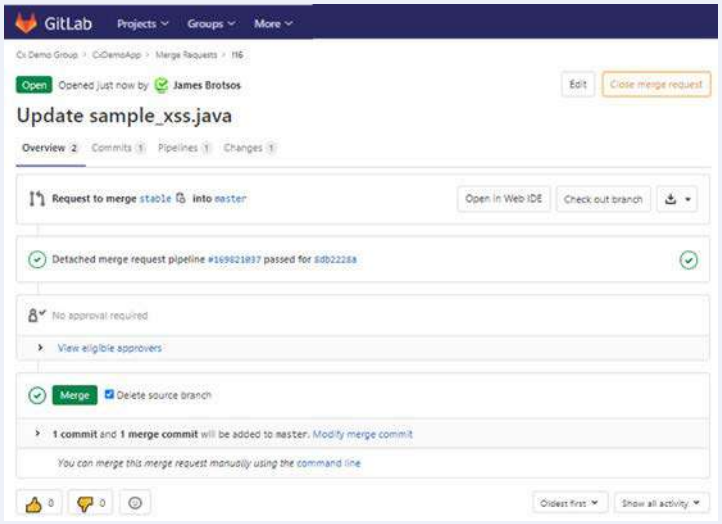
於自動化中更容易實現 Shift-Left

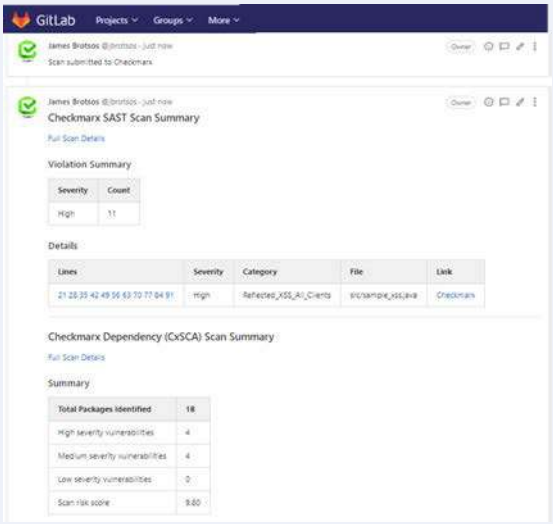
一般來說，市面上常見 CI 如 Jenkins、CircleCI、Travis CI、Bamboo、TeamCity 等，皆支援外掛程式結合 SAST 掃描工具方案，不過在觸發條件的設計通常會在 build 之後進行，而此時檢測的時機於 SDLC 流程中偏晚。

而 CxFlow 有在協作層簡化了實現現代開發環境自動化特性，使得組織得以在功能面上提前進行掃描。

CxFlow 可與 SCM 或是敏捷計畫工具無痛整合，例如 GitHub、GitLab、BitBucket、Azure DevOps 等等，實現完全自動化應用程式掃描與傳遞掃描結果供開發人員查閱。

CxFlow 亦可與 Bug Tracking System 整合，例如 Jira、Rally Software、ServiceNow、SonarQube 等等，減少耗時的人工手動設定，讓開發團隊更輕鬆的追蹤結果。





CxFlow

自動依檢測結果開立 issue 單

利用了 Checkmarx 掃描未編譯程式碼的獨特能力，CxFlow 能在 SDLC 早期自動執行掃描程式碼。大大的降低了對掃描進行手動配置的需要，並允許開發人員根據程式碼管理工具本身的預先配置政策發布和更新掃描結果。

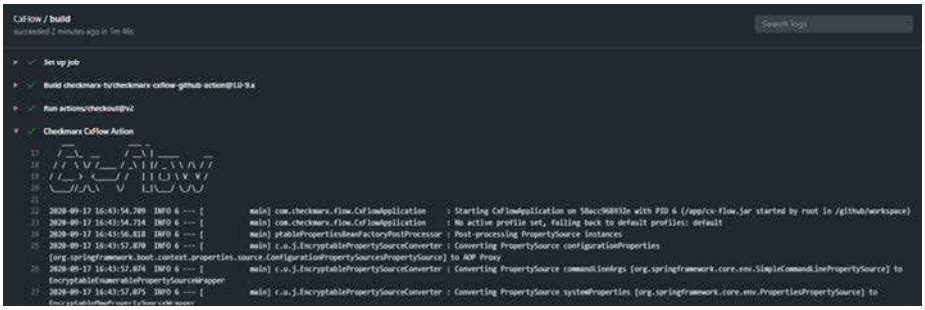
在初始配置之後，SAST 掃描便會自動執行，除了需要開發人員發起的 Pull request 之外，並不需要任何人工作業。

借助 CxFlow，在使用的程式碼管理工具中發出 pull request 時，開發人員不僅會收到管理工具上的留言通知，告知掃描開始，還會收到與安全相關的資訊通知留言，因為 CxFlow 可簡單地與使用中的 IDEs 或程式碼管理工具進行整合。

有鑑如此，開發人員能夠就所有的掃描結果進行一次性的程式碼審查，並能夠使用 Ticketing System 一次性的關閉 Feedback Loop，在他們對程式碼內容還記憶猶新的時候。

使得開發人員得以：

- 在編碼階段（開發最早期階段），進行風險的捕捉與修復。
- 一如往常的作業模式，無需中斷，無需習慣新工具，無額外的安全性檢視等等。
- 整治安全性 Bug 如同修復功能 Bug 一般並使得開發人員直接在目前處理的 Code Branch 中修復錯誤。
- 大量減少手動打開、驗證和關閉安全性 Ticket 單的操作，而且無需在追蹤 Bug 與 Ticket 單上耗費大量時間。



CxFlow

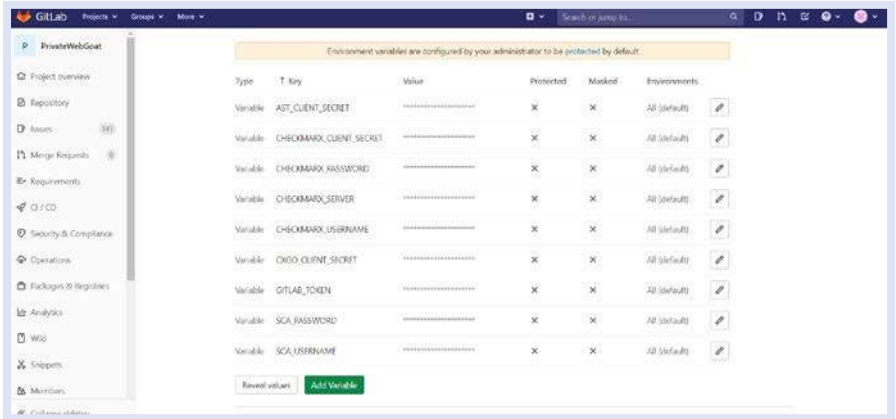
消彌開發人員與 DevOps、AppSec 團隊間的摩擦

CxFlow 減少了 DevOps 中每個 Project 的手動配置與耗時，從而消彌基於將掃描步驟加入到所有 CI pipelines 時與 DevOps 團隊間的摩擦。

- 通過使用 web listener 接收 Source Code repositories 事件，並依據事件觸發掃描動作達成自動化的方法，簡化 SAST 生命週期。
- 開發人員無需熟悉 SAST 工具，直接藉由 Repositories 提供的 comment / reports 功能來取得掃描結果。
- 支援多元分支掃描。例如，您可以配置 master、develop、security branch 等受保護的分支。當這些分支有任何的程式碼變更時，pull request、pull、push 事件都可以觸發掃描並產生結果。
- 當使用多個 CI Solutions 時，通過減少管理和維護多個 CI plugins 的需求，降低 TCO（Total Cost of Ownership） 並提高掃描工具的 ROI。

結論

CxFlow 簡化了開發工具和 Checkmarx 之間的配置與編排，並驅動了 SAST 自動化。有了這個，組織內的開發、安全、維運團隊可即刻到位，並簡化他們的安全策略與 DevSecOps 流程管理。傳統的 SAST Solution 供應商無法直接針對源碼進行掃描，因此未考慮到開發人員，故還在使用傳統工具的人員無法如同使用 Checkmarx 源碼管理工具一般，有高效率的工作品質。顯而易見的，整合是自動化的關鍵，CxFlow 支援 Shift-Left 方法，在這樣的方法中，自動化實際上可以實現在 SDLC 中一改變了 AST solutions 與所有 DevOps 環境的整合方法。



CxFlow

IaC Security 守護雲原生應用程式

近年來，隨著組織上雲需求提高，尋求快速配置基礎設施、橫向擴展方法，由於 IaC 能滿足上述需求，其採用率也大幅上升。然而，隨著 IaC 的優勢，伴隨的即是開發人員也同需面臨安全性、合規性的配置風險。根據「second biggest cause of data breaches.」的研究中也證實了配置不當與錯誤交付是問題的主因。*註1。

Gartner 指出預計在 2025 年，所有企業工作負載的 70% 將部署到雲基礎設施以及平台服務中。而超過 99% 的雲端設施漏洞的根本原因，為事前可先預防的錯誤配置。因此，隨著越來越多的組織利用雲基礎設施並將軟體部署在雲上作為其業務模型的一部分，能夠通過掃描所有環境中的雲基礎設施來不斷降低安全風險與了解其安全狀況是至關重要的。

IaC（Infrastructure As Code）掃描作為軟體開發生命週期的一部分，是開始的第一步，而接下來掃描將進入下一步階段。除了掃描 IaC 文件外，我們還可以連接並掃描已部署的生產環境，以幫助識別這些環境中的任何安全配置錯誤。無論這些錯誤配置來自您的 IaC 文件、手動資源配置和更改，還是資源未與當前版本或安全功能保持同步，Checkmarx KICS 現在都可以幫助解決其中的許多問題。

快速、可擴展的開源 IaC 掃描

KICS（Keeping Infrastructure as Code Secure） 自動解析任何類型的常見 IaC 文件，以檢測可能使您的應用程序、數據或服務遭受攻擊的不安全配置。這意味著您可以讓團隊中的任何人編寫 IaC 文件，然後在發布文件之前檢查這些文件以確保它們是安全的。無需在 IT 治理策略中設置安全指南並希望工程師和開發人員在創建 IaC 文件時遵循這些規則，您可以使用 KICS 自動化檢查 IaC 安全性。此外，由於 KICS 是支持所有主流 IaC 平台（Terraform、CloudFormation、Ansible、Helm 等）並與各種軟體開發工具整合的開源工具，因此可以將 IaC 安全掃描添加到您現有的工作流程中沒有摩擦。現在，您的開發人員不必放慢速度來確保 IaC 安全性。

KICS 支援雲端平台設定

Terraform　kubernetes　docker

AWS CloudFormation　ANSIBLE　OPENAPI INITIATIVE

HELM The Package Manager for Kubernetes

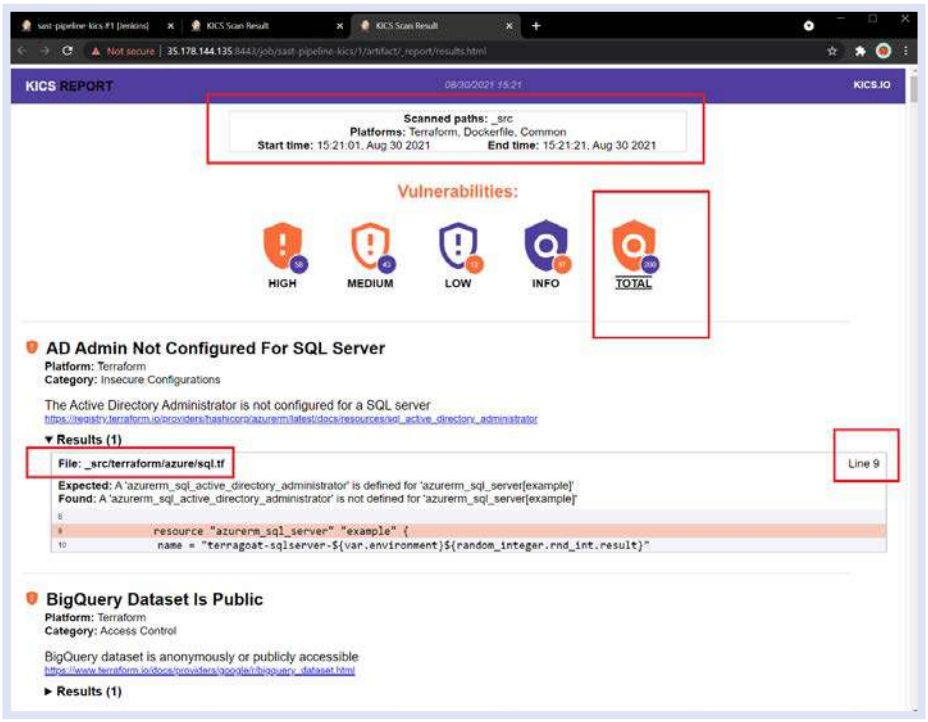
註 1
https://checkmarx.com/press-releases/checkmarx-launches-infrastructure-as-code-scanning-solution-to-secure-cloud-native-applications/
https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf

實施 API 設計最佳實踐

KICS 不僅僅是用於保護單個 IaC 文件的工具。它更進一步，評估您的整體 API 設計是否存在錯誤配置，使您能夠識別路徑定義、身份驗證模式和傳輸加密中的風險。

這意味著您可以為您的組織設置 API 安全標準並通過 IaC 掃描強制執行。 KICS 在應用程序構建時自動運行掃描，因此您可以系統地檢查您的 API，而不會減慢您的軟體交付管道。

您可以充分利用 API 並確保它們可以隨著時間的推移而發展以滿足不斷變化的需求，而不會使您的應用程序暴露於 API 安全漏洞。



KICS 掃描報告

IaC Technology	Query Accuracy[1]	Query Coverage[2]	Scanned IaC files	Number of Results	Average Scan Time (s)	Average Project Size (MB)
Terraform	99.7%	46%	1176	709	6.6	33.4
Docker	98.8%	68%	1017	5109	11	0.7
Kubernetes	99.3%	88.7%	6089	21753	7	90
CloudFormation	95%	73%	1769	5343	10.2	4.8
Ansible	100%	54%	3367	1320	23.3	4.1

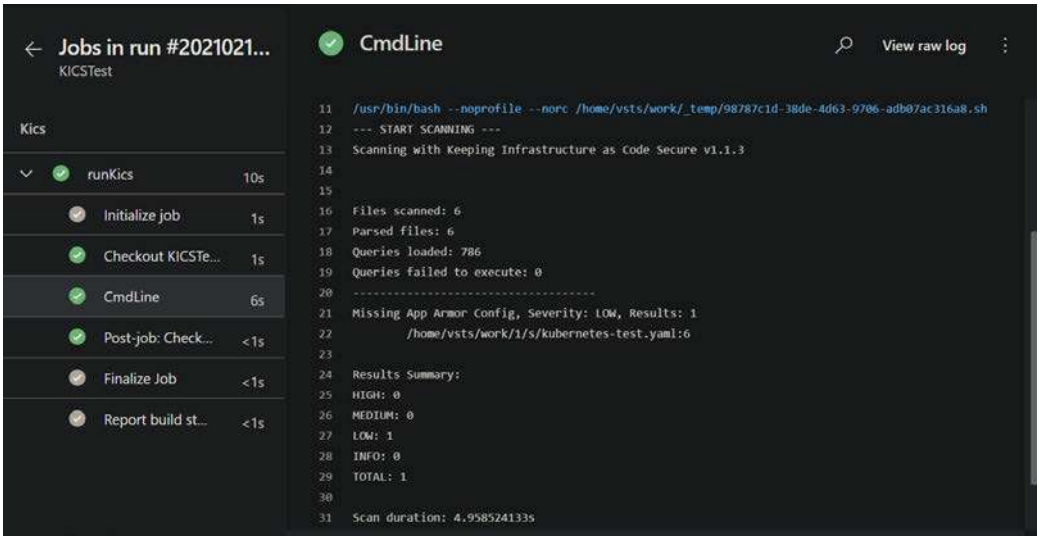
KICS Benchmark

高度可擴展的解決方案

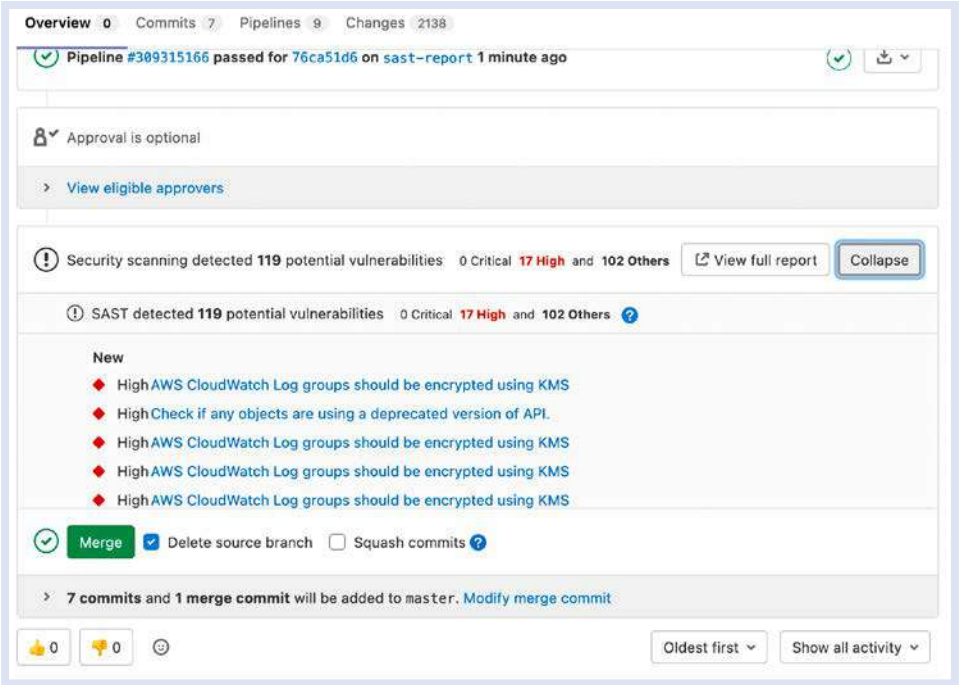
作為與平台無關的 IaC 掃描工具，KICS 可以隨著您的開發和部署操作無縫增長。開發人員可以使用簡單的行業標準查詢語言通過新檢查擴展 KICS。此外，他們可以通過利用 KICS 的模組化設計，將新專案快速加入自動掃描工作流程，同時將 IaC 掃描功能擴展到應用程式、微服務的應用。

KICS 提供靈活、可擴展的解決方案，用於將 IaC 安全掃描整合到您現有的軟體交付週期中。借助 KICS，您可以繼續快速發展並擴大規模，而不必擔心 IaC 文件會在您的環境中傳播安全漏洞。

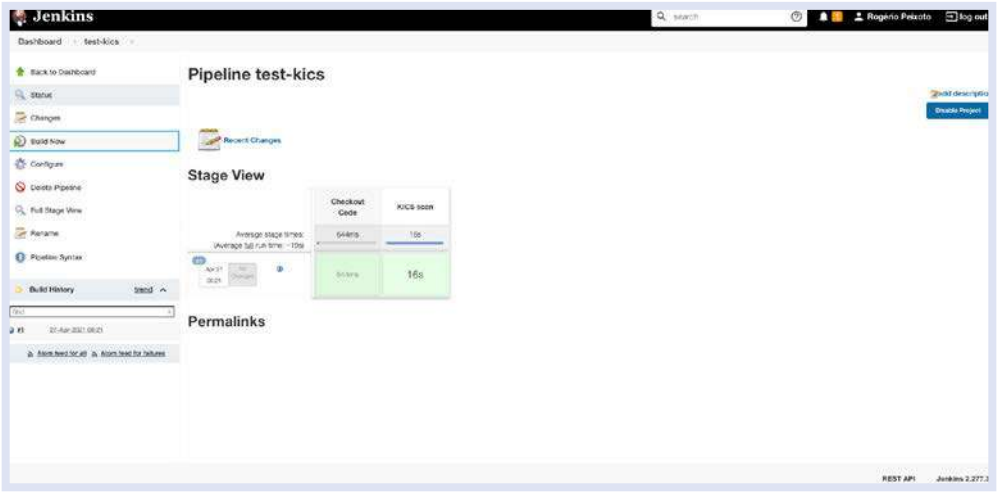
Integration



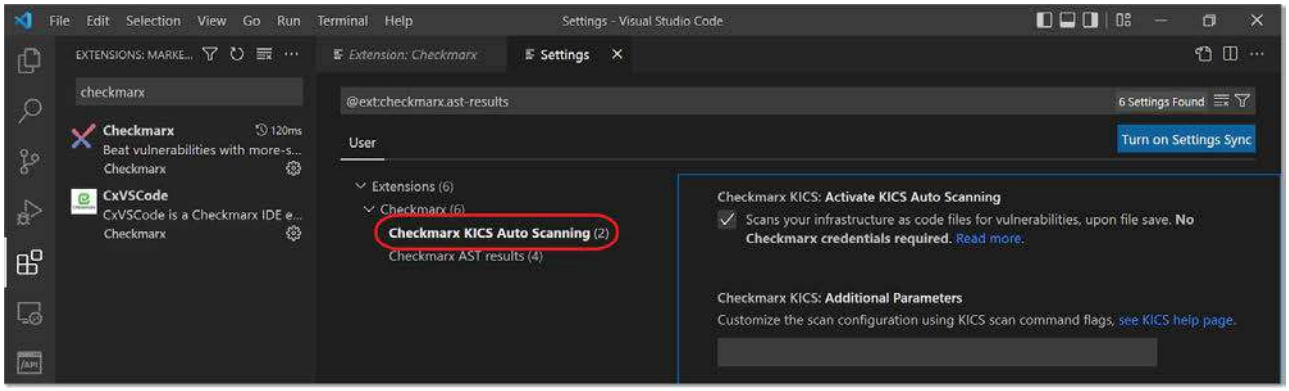
kics_azure_pipelines_success



kics_gitlab_pipeline_sast_report



jenkins-pipeline-success



VS_CODE_Integration

KICS Integration Support

Azure Pipelines	GitLab CI	Terraformer
Bamboo	Jenkins	AWS CodeBuild
Bitbucket Pipelines	TeamCity	KICS Auto Scanning Extension for Visual Studio Code
CircleCI	Travis	
Codefresh	Pre-commit hooks	More soon...
Github Actions	Terraform Cloud	

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

AppSec 開發培訓課程 （CxCodeBashing）

現今快速的開發環境中，需要快速並持續整合、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。

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此目標。

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，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。因此在進行這類活動時，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

多數的安全程式撰寫教育訓練並未確實仿造遊戲的設計原則與元素。然而當學員處在享受的環境中學習，成效或許會更好。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。」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。

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。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

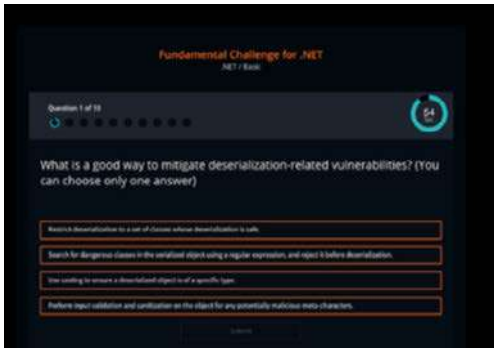
故事背景介紹、角色扮演可以刺激學員的反應，當現實生活中若面臨相同情況，將需如何應對？安全程式撰寫教育訓練若以條列式的問題、答案易讓學員產生例行公式的厭煩感。而情況、人物、所面臨的問題可避免這個問題。依據故事情節（攻防實例、需解決的漏洞），有助於記住所學的內容。而故事也是許多遊戲的精髓，往往是一種有趣的催化劑，與遊戲化是切不可分的。

Ensure They Win 確保參與者贏

依據 Dr. Ian Robertson 著名的研究「The Winner Effect」。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。

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。保持資訊精簡是最好的方式。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。



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。

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

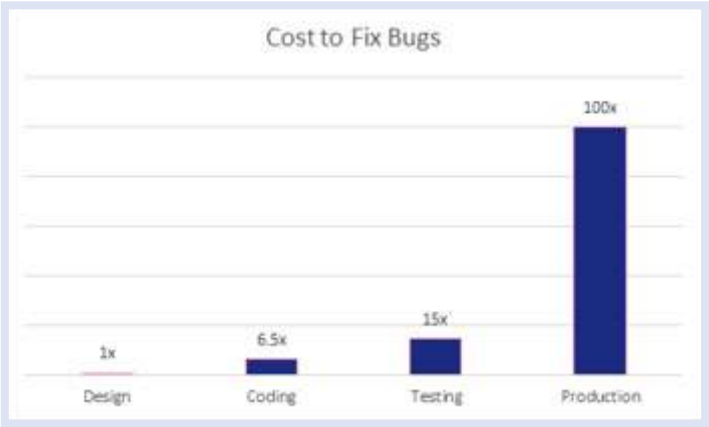
支援語言

Checkmarx ONE

最大化應用程式安全

在外界環境變化快速的 AppSec 中，資安人員爭先恐後地跟上最新的解決方案、工具、報告、分析和新聞的情況並不罕見，您可能正在安排、關聯和組合來自 SAST、SCA、Container 或基礎架構即程式碼 （IaC）掃描引擎的掃描結果，或者查看有關新發現的 CWE 或 Zero-Day 攻擊。如果您是開發人員，您可能會在程式碼中進行分類、確定優先順序和應用修復程序、版本更新、與安全和業務團隊一起審查掃描結果，並在修補過程中或新版本發布之前與測試人員和 QA 協作。

使用過度分散的工具、掃描、報告、自動化技術和整合，在不暴露敏感訊息和組織信譽的風險的情況下，持續提供業務價值可能會非常耗時且困難。再加上平均而言，組織在線上營運環境修復錯誤的成本是設計階段成本的 100 倍，而且很明顯，在軟體開發生命週期 （SDLC）的早期漏洞識別和修復是關鍵。



雲端時代應用安全挑戰

利用應用程式安全測試 （AST）平台可以降低成本並使 AppSec 和開發團隊能夠交付高質量、安全的軟體，同時最大限度地降低業務成本和延遲。

88%	86%	30%	45%	150:1
經歷過資訊洩漏平均支出 $4.4M 美元	已知有風險的程式碼被部署	的 CISOs 指出缺乏足夠的關聯性報表，呈現完整的風險態勢	的組織在 2005 年會經歷過 Supply Chain 攻擊	開發人員與資安專業人員比例

在過去 12 個月中，88% 的組織至少經歷過一次資訊洩露事件，平均造成 440 萬美元的損失。而有 86% 的 AppSec 經理和開發人員表示，為了符合上線期限，仍選擇部署已知有風險的程式碼，因為安全性不應拖延企業的運營。30% 的資安長並未有足夠的資訊了解其應用程式安全型態。到 2025 年，將會有將近一半的企業恐遭受供應鏈攻擊。而截至目前開發人員與資安人員的比例為 150：1，大量地缺乏資安專業人員。

AST 平台特色

- 使用 AST 平台而不是 AppSec 的分段或分段點解決方案，AST 平台可以最大限度地減少和消除間接成本，使你的團隊能夠更專注於核心競爭力，而不是輔助或繁瑣的操作任務來增加的工作項目。
- 更新、維護、修補和備份 IT 基礎設施或安全軟體，整理和組合安全掃描結果以進行統一報告等簡單的事情可能會給組織帶來巨大的成本，並分散其人員和業務的核心使命。
- 借助 AST 解決方案基於多次掃描結果整理、關聯和自動化數據的能力，組織可以更快地保護其應用程序並通過代理其業務。

Checkmarx One 簡介

Checkmarx One ™ 是當今最全面、最受行業信賴的 AST 平台。只需單擊一下，您就可以觸發 SAST、SCA、IaC、Container 和 API 安全掃描，自動關聯多個掃描結果並確定其優先順序，以便於使用和確定優先級，並提出或解決風險。此外，掃描可以在您的 SCM 中的程式碼推送事件或拉取請求期間自動觸發，從而無需各個開發人員壓縮他們的工作並上傳到一個或多個掃描工具。 Checkmarx One 不斷創新並在平台中引入新的特性和功能。

Checkmarx One 資源

資安團隊

- 易於檢視的合規報告
- 有效的降低授權費用
- 降低營運開銷

企業

- 加快新的差異化功能的上市速度
- 改善組織風險變化並提高對漏洞和整體攻擊面的可見性
- 保障信譽

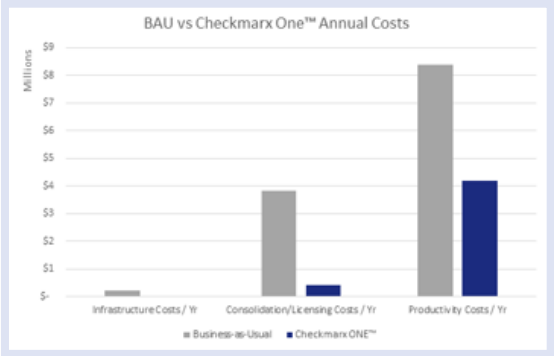
開發人員和開發運行團隊

- 採用 Shift–Left 方法，在 QA 或 Production 階段進行修復，降低 10~100 倍的成本
- 通過自動分類、關聯和識別源程式碼中的最佳修復位置（BFL）來縮短補救時間
- 在 CI/CD pipeline 中自動掃描和整合，減少開銷
- IDE 整合多個掃描結果和源程式碼識別，讓開發人員在最熟悉的工具中使用
- 整合遊戲化安全培訓解決方案，幫助開發人員從一開始就學習如何編寫更安全的程式碼

Checkmarx One 結論

由於 Checkmarx One 是一種 SaaS 解決方案，您可以毫不費力地獲得功能和平台升級 –– 無需額外的 IT 基礎設施、安裝或修補程序，或處理資料庫日誌。

與傳統 AppSec 解決方案相比，每年可節省 60%，投資回報率為一年或更短，無論是在減少授權或基礎設施成本，還是通過遷移到 Checkmarx One 提高生產力上。

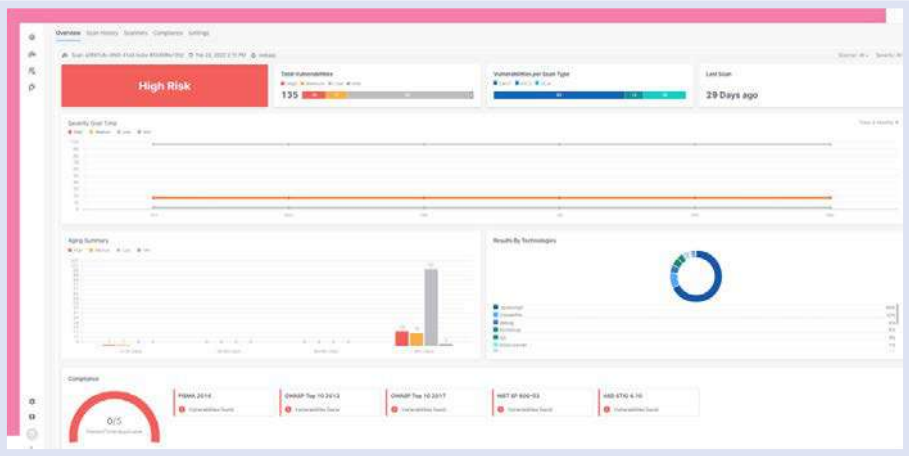


業界最全面的應用程式安全平台

靜態應用系統安全測試工具（SAST，白箱）	在軟體開發過程中提早並經常性掃描程式碼來識別應用程式碼中的漏洞，提供如何在程式碼級別修復複雜安全問題的建議。
軟體組合分析（SCA）	為您和團隊提供所需的工具和洞察力，以解決創建、部署和維護的應用程式中的開源程式碼相關的漏洞和授權風險。
供應鏈安全（SCS）	使開發人員能夠對相依項目執行漏洞、行為和聲譽分析，為您提供更全面的方法來防止供應鏈攻擊和保護開源使用。
API Security	定期檢查和可操作數據，在 API 上線前保護免受漏洞以及任何暴露在應用程式和敏感數據上。
DAST（黑箱）	透過由外到內檢測應用程式，模擬攻擊者發起各種攻擊場景，在運行時發現未識別的漏洞，對正在運行的應用程式進行額外的安全分析。
KICS（IaC）	掃描您的 IaC 文件以查找安全漏洞、合規性問題和基礎設施配置錯誤。通過 2,000 多個預定義查詢，KICS 可以幫助您在部署基礎設施之前快速找到 IaC 安全問題。
Container Security	提供有關基於容器的系統和工作負載的當前安全狀態的信息，包括容器映像和正在運行的容器。

Checkmarx One 最佳選擇

- Aggregated scans：從單個操作觸發多種掃描類型並關聯結果以獲得完整、更準確的程式碼安全圖。
- Faster time to value：通過快速啟動、簡單的配置和增強的掃描調整，在幾小時而不是幾天內啟動您的 AppSec 程式。
- Speed and scalability：以需要的任何容量利用安全的雲驅動掃描，無需管理掃描基礎設施。
- Lower friction and overhead：將該平台整合到您現有的軟體構建 pipeline 和 feedback 系統中，而不是使用會減慢軟體開發和交付速度的獨立 AST 解決方案。
- Wide technology coverage：涵蓋了 30 多種語言、最流行的包管理器和不斷增加的 IaC 模板列表。



支援語言與環境

TABLE 1: SUPPORTED LANGUAGES AND ENVIRONMENTS

SAST Supported Languages		SCA and Package Managers Language Support		IaC Format Support
Android	Kotlin	Java	Microsoft .NET	Ansible
Apex	Microsoft .NET	JavaScript	Node.js	Terraform
ASP	Perl	React	PHP	AWS CloudFormation
C++	PHP	Angular	Python	AWS Cloud Development Kit
Enterprise Cobol for z/OS	PL/SQL	C#	Scala	
	Python	F#	TypeScript	AWS SAM
Go	Ruby	Groovy	WCF	Docker
Groovy	Scala	Kotlin	WPF	Google Deployment Manager
HTML5	VBScript	**Package Managers**		
iOS	Visual Basic	Bower	NuGet	gRPC
Java		Composer	Pip	Helm
JavaScript		Gradle	SBT	Kubernetes
		Maven	Yarn	Microsoft ARM
		NPM		OpenAPI 3.0

Checkmarx 保障您的資料安全

安全、分割、可用性

- 數據安全與分隔

大多數組織認為他們的應用程式程式碼高度敏感，Checkmarx One 從一開始設計了應用程式安全平台，具有數據加密、最少的數據保留和用戶之間強大的數據分隔。

Table 1: Data Security

Component	Controls
Data in transit	TLS 1.2 with ECDHE cipher preferred
Data at rest	AES-256 encrypted
Data segmentation	Dedicated tenant storage buckets
Data retention	Full project source retained for six months; metadata until end of contract
Platform authentication	User/Password with two-factor authentication and identity federation
Role-based access control	A granular set of permissions as well as robust group and role management for flexible user access management