

OpenText Static Application Security Testing

找出原始碼安全性弱點的根本原因，優先處理最嚴重的問題，並取得詳細指引瞭解如何修正

整合共生體系包括：

- 彈性的部署選項：AppSec 即服務、內部部署或雲端。
- 整合式開發環境 (IDE)：Eclipse、Visual Studio、JetBrains (包括 IntelliJ)。
- CI/CD 工具：Jenkins、Bamboo、Visual Studio、Gradle、Make、Azure DevOps、GitHub、GitLab、Maven、MSBuild。
- 問題追蹤器：Bugzilla、Jira、OpenText Software Delivery Management。
- 開放原始碼安全管理：Sonatype、Snyk、WhiteSource、BlackDuck。
- 程式碼儲存庫：GitHub、Bitbucket。
- Swaggerised API 可無限制自訂。
- 便於開發人員使用的語言涵蓋範圍：
 - 支援 Java、Kotlin、Scala、C#、VB.NET、TypeScript、JavaScript、C/C++、Python、PHP、Go、COBOL、Swift、Objective C/C++、Salesforce Apex、Dart/Flutter、Bicep、Solidity、Ruby, SAP ABAP、PL/SQL、T-SQL、ColdFusion、ActionScript、Visual Basic 6、VBScript、Ruby、HTML、XML、JSON、YAML、HCL。有關支援的語言詳細資料，請參閱軟體系統要求 [說明文件](#)。

使用靜態測試建立更完善的程式碼

Static Application Security Testing (SAST) 能在修正成本最低的開發初期階段識別安全性弱點。它能針對程式碼在開發過程中所發生的問題，立即向開發人員提供意見回饋，以降低應用程式的安全風險。SAST 也能協助在開發人員工作時指導他們安全性課題，讓他們能建立更安全的軟體。

OpenText™ Static Application Security Testing 使用多種演算法和廣泛的安全編碼規則知識庫，分析應用程式原始碼中可利用的弱點。

此技術會分析執行與資料可遵循的每一可行路徑，以識別並矯正弱點。

及早發現安全性問題

為處理程式碼，OpenText Static Application Security Testing 的運作方式與編譯器類似，它會讀取原始碼檔案，然後將檔案轉換為針對安全性分析進行強化的中繼結構。這種中繼格式用於找出安全性弱點。分析引擎由多個專門的分析器組成，使用安全編碼規則來分析程式碼基底中是否存在違反安全編碼實務的情況。

以 Software Security Center 管理結果

Software Security Center 是集中式管理儲存庫，提供企業的整個應用程式安全計畫可見度，以協助解決跨軟體產品組合中的安全性弱點。使用者可透過管理儀表板和報告來檢閱、稽核、排定優先程度、管理矯正工作、追蹤軟體安全性測試活動，並測量改善的成果，將靜態、動態和軟體構成分析結果最佳化。

OpenText Static Application Security Testing 會持續建立掃描結果與評估結果的關聯性並加以追蹤，讓開發人員透過 Audit Workbench 或如 Eclipse、Microsoft Visual Studio 等 IDE 外掛程式取得資訊。

使用者也可以手動或自動將問題推送至瑕疵追蹤系統，包括 OpenText Software Delivery Management、Jira、Azure DevOps Server 和 Bugzilla。

- Audit Workbench
 - Smart View — 視覺化讓稽核與修正更容易：
 - 從資料流的觀點，快速瞭解多個問題的關聯性
 - 套用 Smart View 過濾器即可開始分類排序，或在最有效率的時間點修正問題

- 整合至 CI/CD 工具 (IDE、錯誤追蹤器、開放原始碼)。
- 支援所有主要 IDE：Eclipse、Visual Studio、JetBrains、包括 IntelliJ。
- 瑕疵管理整合可讓您全盤掌握安全性問題，對症進行修復工作。
- 開放原始碼安全性整合：Sonatype 和 Debricked。
- 結合以 Swagger 支援的 REST API、開放原始碼 GitHub 儲存庫和 Bamboo、Azure DevOps 和 Jenkins 的外掛程式和擴充功能等工具類型，進行 CI/CD 管線自動化。

獲得快速又準確的掃描

Static Application Security Testing (SAST) 會在開發初期擷取大部分與程式碼相關的問題，讓您能夠識別並排除原始碼、二進位或位元組程式碼中的弱點。OpenText 可偵測超過 33 種程式設計語言的 1,627 個獨特弱點類別，並涵蓋超過一百萬種個別 API，其準確度獲得 OWASP 1.2b Benchmark 中 100% 真陽性率的證實。

將 CI/CD 管線中的安全性自動化

OpenText 可透過識別及排序哪些弱點構成最大威脅來降低風險，並整合 CI/CD 工具，包括 Jenkins、OpenText Software Delivery Management、Jira、Atlassian Bamboo、Azure DevOps、Eclipse 和 Microsoft Visual Studio (請參閱 [OpenText 整合](#))。即時檢查掃描結果，並提供建議與程式碼行號瀏覽，以更快找到弱點並執行協同稽核。

縮短開發時間與節省成本

內嵌於 SDLC 內時，開發時間與成本可減少達 25%。在上線/發行後階段發現的弱點，修正成本是生命週期早期發現之弱點的 30 倍。OpenText 讓開發人員在工作時瞭解靜態應用程式安全測試，以實現安全編碼實務。

選擇彈性的部署選項，以配合您團隊所在的開發環境：

- OpenText Core Application Security 可讓團隊在完全以 SaaS 為基礎的環境中工作。
- Hosted 以獨立的虛擬環境運作，完全控管使用者資料，讓您享有 SaaS (軟體即服務) 與內部部署兩者的最佳效益。
- On-Premises 可讓團隊絕對掌控 OpenText 解決方案的所有層面。

開發人員可取得即時安全性分析和結果

Security Assistant 提供為提高速度和效率而打造的結構和組態分析器，以支援我們最即時的安全性回饋工具。它使用 IDE (Microsoft Visual Studio、Eclipse 和 IntelliJ) 立即產生的結果，只會找到高度信心 (全為真陽性，或極低的偽陽性率) 的結果。

建議使用 OpenText Core Application Security 搭配 Security Assistant 作為開發人員的額外工作輔助，並搭配完整靜態掃描，以更全面檢視安全性問題。所有目前的 OpenText Static Application Security Testing 和 OpenText Core Application Security Static Assessments 客戶都有權使用 Security Assistant，無需額外的授權/成本。

「OpenText 讓我們
以更敏捷快速的方式
分析更大量的程式碼。
現在我們的管線通常
不會發生弱點錯誤，
因為這些錯誤早在
開發程序就已偵測
到了。」

Location World
DevOps 主管
Wilson González

減少手動稽核時間

Audit Assistant 利用機器學習節省手動稽核時間，找出貴組織最相關的弱點並訂定優先順序。利用套用的機器學習進行自動化，可縮短手動稽核時間，提高靜態應用程式安全測試計畫的投資報酬率。Audit Assistant：

- 在數分鐘內提供自動化稽核結果。
- 盡可能降低稽核員的工作負擔。
- 以信心程度排定問題的優先順序。
- 為整個專案建立準確且一致的稽核結果。
- 配合 DevOps 的速度交付結果。
- 減少需要深入手動檢查的問題數量。
- 更快找出相關問題並排除偽陽性。
- 利用現有資源擴充應用程式安全性。

取得集中式掃描基礎架構

ScanCentral 可在組建伺服器上實現輕量封裝，並從 Software Security Center 內提供集中化的掃描基礎架構，以滿足不斷成長的現代開發需求。其可透過內部部署、隨選或混合方式進行擴充。ScanCentral 可透過調整掃描和改善掃描效能，提供達成所需涵蓋範圍的彈性；調整為快速掃描；並微調為完整且更準確的 Restful API/Swaggerized API。