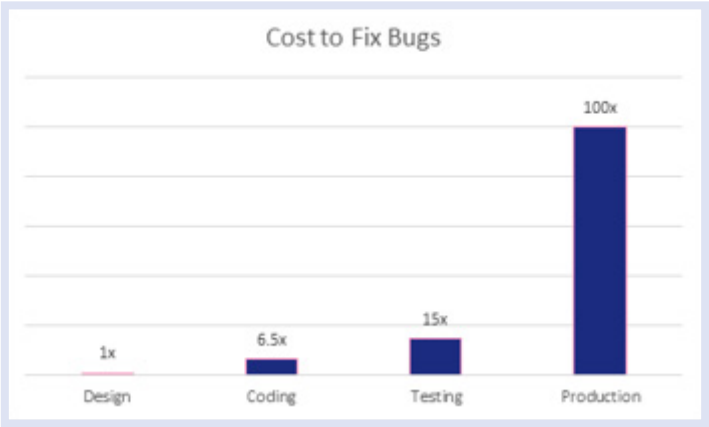
Checkmarx ONE

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

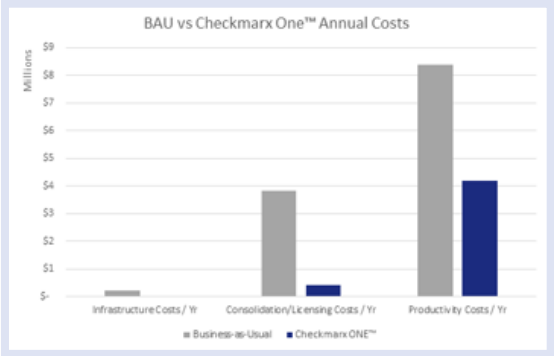
開發人員和開發運行團隊

- 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

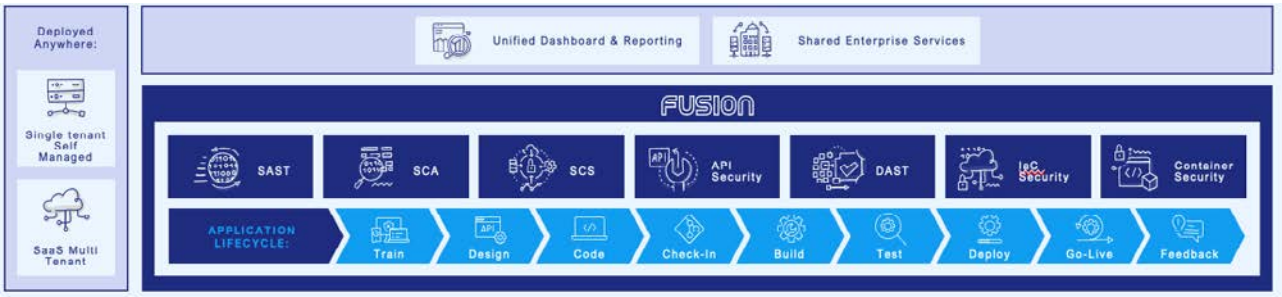
Checkmarx One 結論

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。



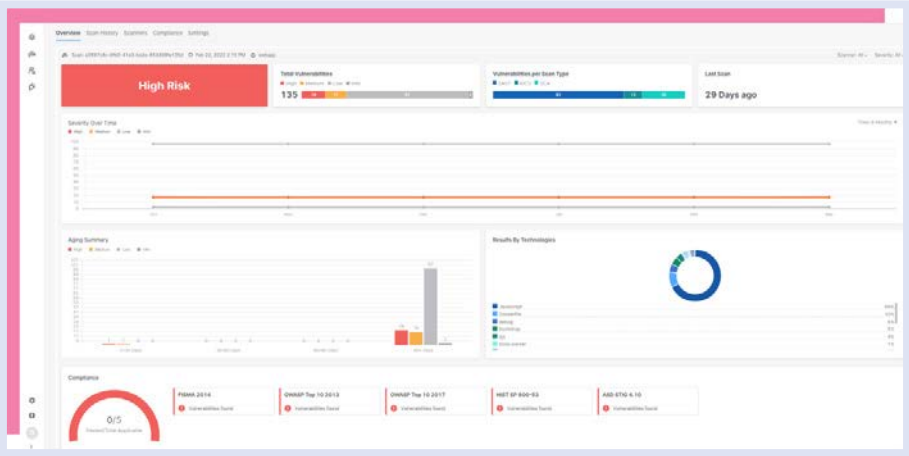
業界最全面的應用程式安全平台



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

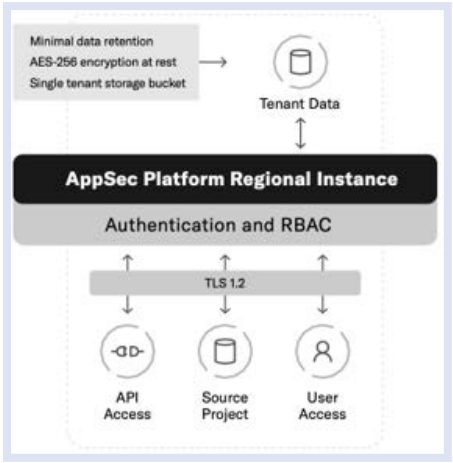
安全、分割、可用性

- 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



Checkmarx
The world runs on code. We secure it.

應用系統測結果關聯性 (Correlation) 在現代應用系統開發中 為何如此重要

全方位視角的優劣與驅動因素

軟體是數位轉型的核心，企業肩負著龐大壓力滿足客戶的需求，完成各式複雜的架構，提供更快、更聰明且容易使用的應用系統。有鑑於此企業必須針對現代應用程式開發（Modern Application Development, MAD）的應用系統安全測試（AST）解決方案中，獲得經過關聯分析的掃描結果。

應用系統安全測試關聯結果不可或缺的原因如下：

1 跟上雲端時代

長期保持軟體安全至關重要，因此企業正在轉向採用全面性整合的多合一應用系統安全測試（AST）平台，以跟上雲端時代的步伐。從掃描所有程式碼到結果分析，Checkmarx One AST PlatformTM 能更快得出結果，並提供更好的分析資訊與降低風險。

此外，我們最近的調查發現，「上雲」是現代應用程式開發面臨的最普遍的挑戰，其次則是雲端安全（54%），以及採用雲原生安全測試（37%）。

2 以驚人的速度運作

現代應用系統一切皆自動化，這意味著將不再有人為因素造成延遲，企業可以快速交付軟體，也能在短短一天內產出數個新版本。但若缺乏關聯式掃描結果，一個微小錯誤可能導致星火燎原，帶來毀滅性的衝擊。

事實上，現代應用程式開發已經能讓原本為期五年的軟體專案縮短至一年 — 這可是高達 400% 的技術成長！

3 安全部署程式碼

隨著部署在雲端的應用系統逐漸增加，組成這些應用系統的原始碼、微服務、Containers、開源程式碼和基礎架構程式碼，也在現代應用系統開發下產生新的漏洞、風險與暴露程度。有鑑於此，企業需要經過相互關聯的掃描結果才能捍衛日益增加的受攻擊面。

當被問到現代應用程式開發應優先考慮的領域時，最普遍的回答是：Containers（31%）、Serverless（32%）、基礎架構程式碼 (IaC)（28%）和混合雲（28%）。

Checkmarx Fusion™ 簡介

Checkmarx 推出了 Fusion，為現代應用程式環境中帶來進階關聯式功能。透過關聯 Checkmarx One AST 平台中多個應用系統安全（AppSec）引擎的掃描結果，您將透徹理解各種潛在漏洞的真正影響。

可視性

提供威脅模型分析，透過直觀且容易理解的圖表描繪所有軟體元件及其使用的雲端資源，圖象呈現與潛在威脅的關係。

關聯式

將靜態與動態檢測結果進行整合與關聯，比起各自單獨的掃描更能因結合了不同層面的方式，幫助優化檢測結果減少誤報。

優先順序

根據漏洞的實際風險和潛在影響，評估漏洞的優先順序，讓團隊聚焦於最迫在眉睫的危機。

雲原生

能涵蓋雲原生架構，包括微服務、雲端資源、Containers 和 API，同時也將預備部署到執行期的洞察報告進行關聯。

「Fusion 在單一平台上彙整相關掃描結果，並提供關聯讓人一目了然 — 這與市面上僅將各種資料陳列在一起的其他解決方案完全不同。」

– Mark Mishaev，Checkmarx 研發部門首席工程師

CHECKMARX FUSION

藉由直觀且全面的圖表呈現漏洞與風險，以優化應用系統安全

孤立的安全測試工具在現代應用系統開發（MAD）過程遭遇的艱鉅挑戰

解決方案簡介

和過往由單一程式語言編寫的單體應用程式不同，當今的一個雲原生應用程式，就可能包含多達 50~5000 個不同的組件，而其中任何一個組件都可能隱含讓惡意攻擊者有機可乘的漏洞。鑑於現代應用系統的規模及複雜程度，應用系統安全測試也應與時俱進滿足現況。

以下是常見的應用系統安全測試方法：

- 靜態應用系統安全測試（SAST，白箱）
- 軟體組成分析（SCA）
- 動態應用系統安全測試（DAST，黑箱）
- 互動式應用系統安全測試（IAST，灰箱）
- 基礎架構程式碼掃描（Infrastructure as Code，IaC）

然而上述工具皆只聚焦整體架構的特定區塊，而各自獨立運作而在現代應用系統開發（Modern Application Development，MAD）中屢次被驗證效率不足。其中包含以下原因：

1）結果未整合：個別軟體本身或許沒有漏洞，不過一旦將其整合，可能引發新的漏洞與風險，而這些漏洞絕非單一掃描程式所能檢測。即使公司購買同類產品中最頂尖的掃描程式，當它們單獨運作時，也無法在現代應用系統開發（MAD）環境中針對潛在漏洞及風險提供全方位視角與完整的流程，這將埋下安全問題的隱憂。

2）時間點不同：每種檢測工具都僅針對特定的漏洞，因此若檢測工具被部署在軟體開發生命週期的不同階段，則會得到完全不同的結果。

由於分析應用程式原始碼的靜態應用系統安全測試（白箱）工具，經常被部署在軟體開發生命週期的開發及測試階段，這個可能會因為沒有與 Run-time 結果整合，而得到不少誤報。但若於在 UAT 或正式環境階段，針對部署後的應用系統進行動態應用系統安全測試（黑箱），雖然可以發現許多漏洞，但當下往往已經沒有時間修復。

由於誤報與最後一刻才發現的漏洞，缺乏關聯的結果，讓開發人員和資安團隊身陷人工比對、手動整合及分析資料的繁瑣作業中，被大量且非關聯式的檢測結果淹沒。

最後，因為無法充滿信心地分類並修復漏洞，且亦缺乏對應用系統安全狀況的全方位視角，公司只能延後產品發表時程，或是發表不安全的軟體 — 如此不僅傷害自己的商譽，也損及客戶的事業與資料。

Checkmarx Fusion ™ 簡介

源於手動分類的效率低落，而常見的替代解決方案僅是將無關的結果放在一起，因此 Checkmarx 開發了 Fusion，這是一項為客戶提供進階關聯式功能的解決方案，以迎戰現代應用系統開發（MAD）環境。

Checkmarx Fusion 將來自 Checkmarx One AST Platform ™將多個應用系統安全引擎的測試結果進行彙整與關聯難，以分析當前漏洞和風險的真正影響。網羅這些無法由單個引擎自行發現的漏洞和風險後，Fusion 便能產出有意義、排定優先順序且可執行的分析結果報告。

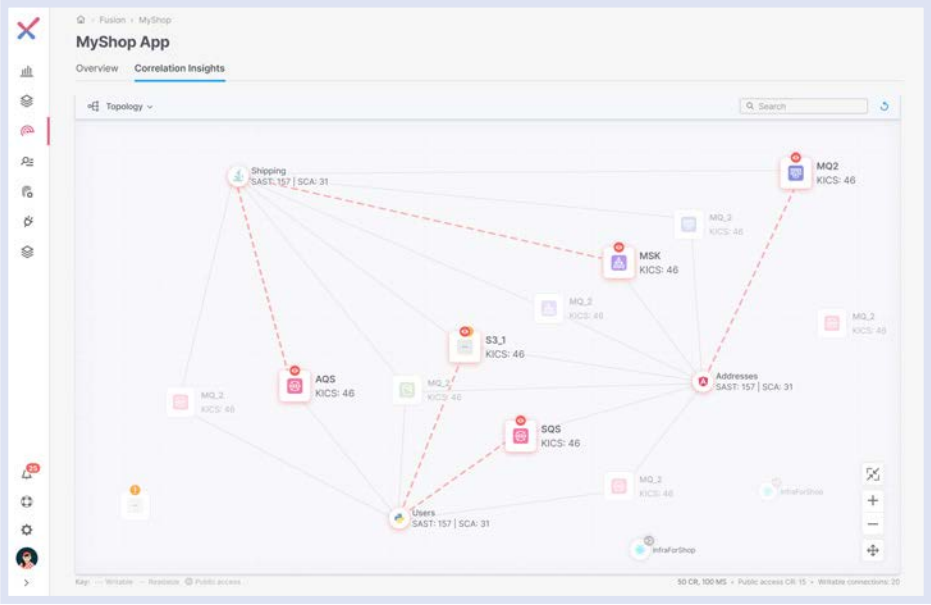
Checkmarx One 應用系統安全測試平台

Fusion 是 Checkmarx One 應用系統安全測試平台的一環，可適用於軟體開發生命周期中的開發與部署階段，其執行靜態與動態的分析可橫跨數個應用系統服務與儲存庫。Checkmarx Fusion 透過全面分析所有漏洞，將其導致的整體影響提供給使用者，Fusion 能為您帶來：

- 可視度：提供威脅模型分析，透過直觀且容易理解的圖表描繪所有軟體元件及其使用的雲端資源，圖象呈現與潛在威脅的關係。
- 關聯式：將靜態與動態檢測結果進行整合與關聯，比起各自單獨的掃描更能因結合了不同層面的方式，幫助優化檢測結果減少誤報。
- 優先順序：根據漏洞的實際風險和潛在影響，評估漏洞的優先順序，讓團隊聚焦於最迫在眉睫的危機。
- 雲原生：能涵蓋雲原生的架構，包括微服務、雲端資源、Containers 和 API，同時也將預備部署到執行期的洞察報告進行關聯。

描繪威脅：拓撲圖表

Checkmarx Fusion 能根據常見的雲原生應用程式的串接與服務，描繪不同引擎掃描的結果。以下是關聯式功能的幾個範例：

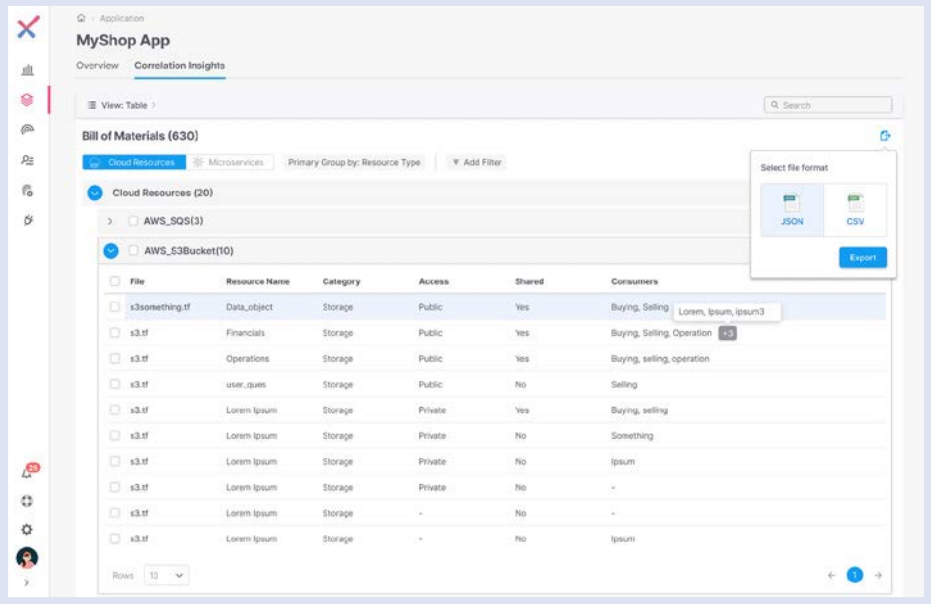


- **靜態應用系統安全測試工具（SAST，白箱）與 KICS[1]：**敏感資料寫入公開的 Amazon S3。
- **靜態應用系統安全測試工具（SAST，白箱）、KICS 和 API 安全工具：**由靜態應用系統安全測試工具（白箱）發現的 API 端點，其中有透過 KICS 發現的安全設置缺失，並由 API 安全工具補充同一端點的詳細資訊。
- **軟體組合分析（SCA）和 KICS：**發現開啟特定埠號 (Port) 時，有機會引發已公開 CVE 編號的 Code Injection 風險。
- **靜態應用系統安全測試工具（SAST，白箱）與靜態應用系統安全測試工具（SAST，白箱）：**微服務間彼此傳遞敏感性資訊，而接收端的微服務以明文方式儲存這些敏感資料。

KICS1: Checkmarx 的 KICS（Keeping Infrastructure as Code Secure ）是一個免費的開源解決方案，用於基礎架構程式碼（IaC）的靜態程式分析。

Checkmarx Fusion 的實際應用：主要使用場景

了解應用程式的物料清單 BOM(Application's Bill of Materials)



Checkmarx Fusion：列表

Checkmarx Fusion 的另一個實際應用例子，是透過雲端資源的可取用性為公開或是私有，來判斷靜態應用系統安全測試（白箱）結果的優先順序。

其他實例包括特別標示原生雲應用程式中多個微服務使用的共享資源，並根據 API 屬於私有或公開等 API 安全資訊，對靜態應用系統安全測試（白箱）結果進行優先排序。