Microsoft Azure

Secure DevOps：應用安全原則與自動化安全

實現安全合規的自動化流程

模組概述

- 左移測試
- 軟體組成分析
- Lab —— 自動化安全流水線

Microsoft Azure

左移測試

靜態應用測試

原始程式碼掃描應通過實施靜態應用安全測試（SAST）來覆蓋

自動化 SAST

不同構建的分段 SAST

- CI 構建
- PR 安全驗證測試（SVT）構建

SAST 應該是提交主分支的必備條件

代碼審查中的安全

從 SAST 開始

- 編碼錯誤
- 憑證掃描
- 安全問題

尋找

- 自製加密或過時的哈希函數
- API 包裝器的錯誤使用
- 糟糕說明類的利用
- 不當的logging技術

動態應用安全測試

動態應用程式掃描工具（DAST）旨在掃描處於運行狀態的預發佈和生產網站，分析輸入欄位、表單以及應用程式的多個方面，以檢測漏洞

保障您的雲環境部署安全

部署前對配置進行安全驗證測試

最佳實踐是利用基礎設施即代碼（IaC）概念，並在可能的情況下重建

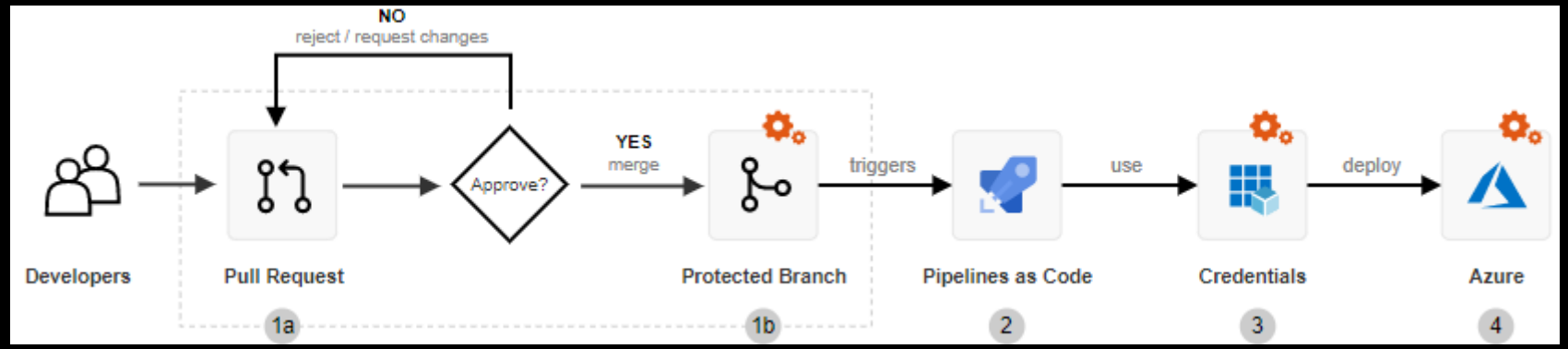
利用 Azure 策略和 Secure DevOps Kit for Azure 進行強制執行和配置漂移

保護您的管道

部署到生產環境應實現全自動化

只有 DevOps 工程師（有限團隊）才應該有許可權編輯生產流水線

利用 Azure DevOps 組織審計功能監控活動



採用分階段方法

採用戰術性和分階段部署可以減輕 Secure DevOps 方法的複雜性

專注於快速獲勝或“唾手可得的果實”。

一旦達到某個特定狀態，就進入下一階段

評估哪些部分可以在開發者的 IDE 中遷移和監控

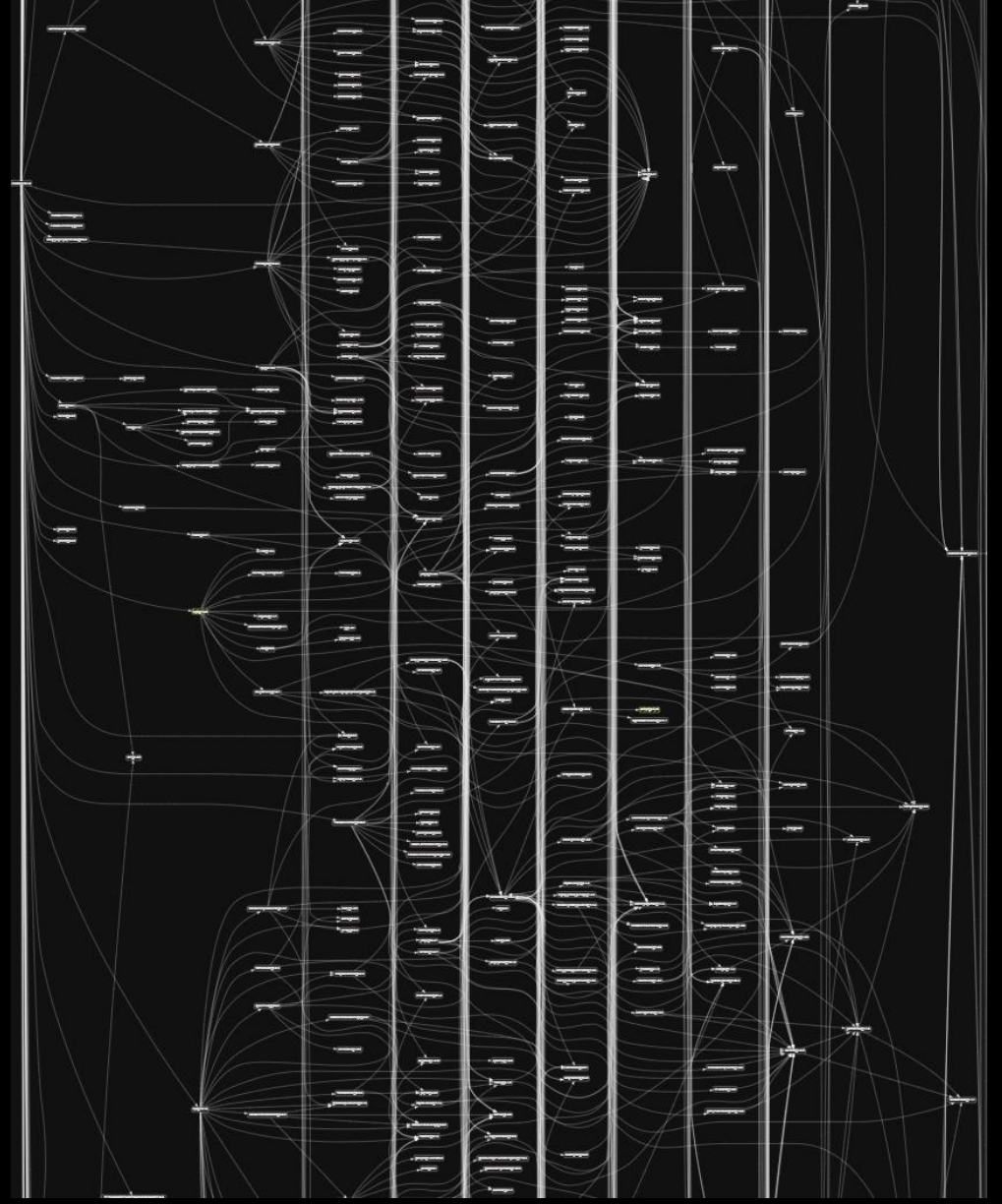
- 在 IDE 中運行憑證掃描
- 在 Azure 的 Secure DevOps Kit 中，及時學習 Security Intellisense
- 在開發者 IDE 中整合一個 linter（提示器）
- 要求開發人員在 IDE 中運行靜態代碼分析器

Microsoft Azure

軟體組成分析（SCA）

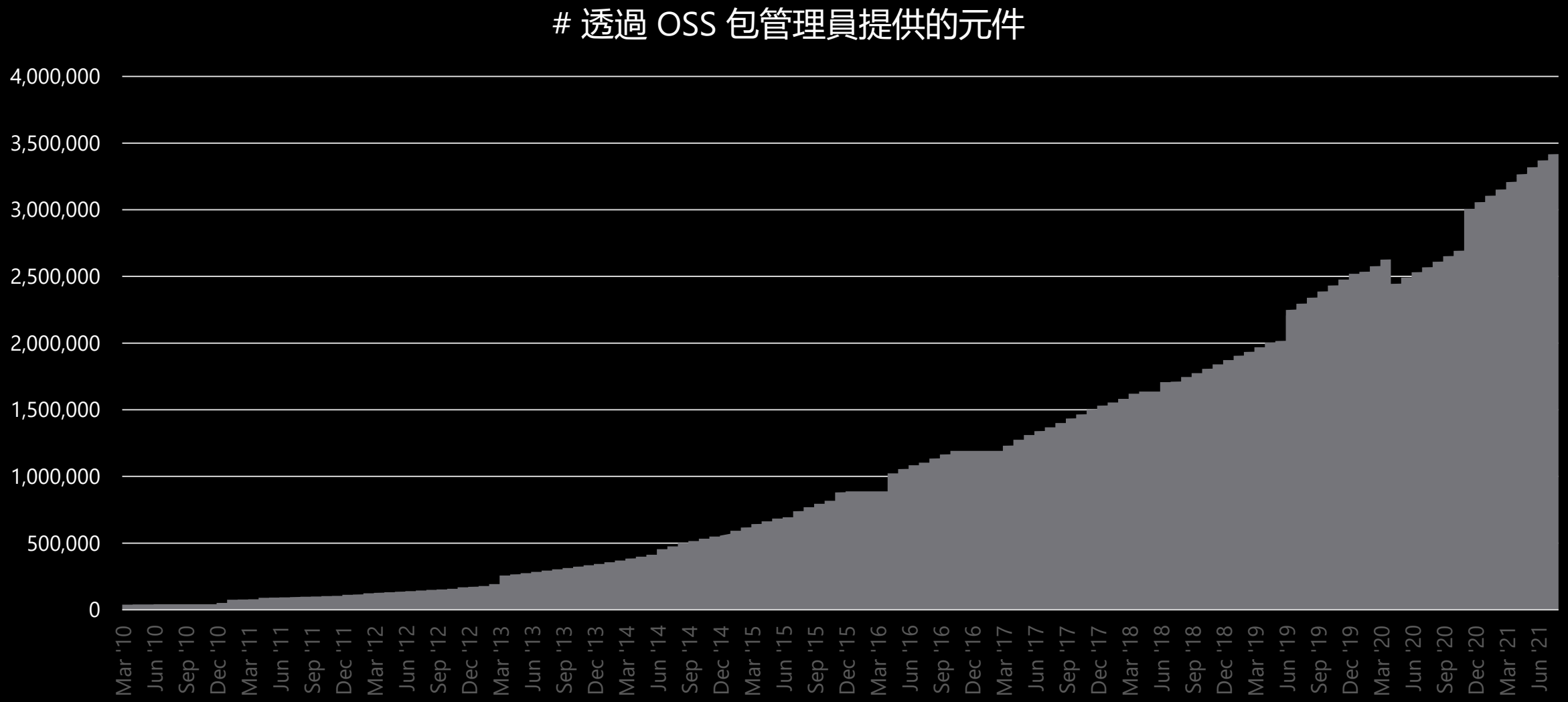
複雜性不斷增加……

《密碼銀河視覺化》（anvaka.github.io）



NPMgraph - 摩卡

SCA —— OSS 套件的數量增長



SCA —— 漏洞數量的增加


每年發佈的 CVE 數量
18,000
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

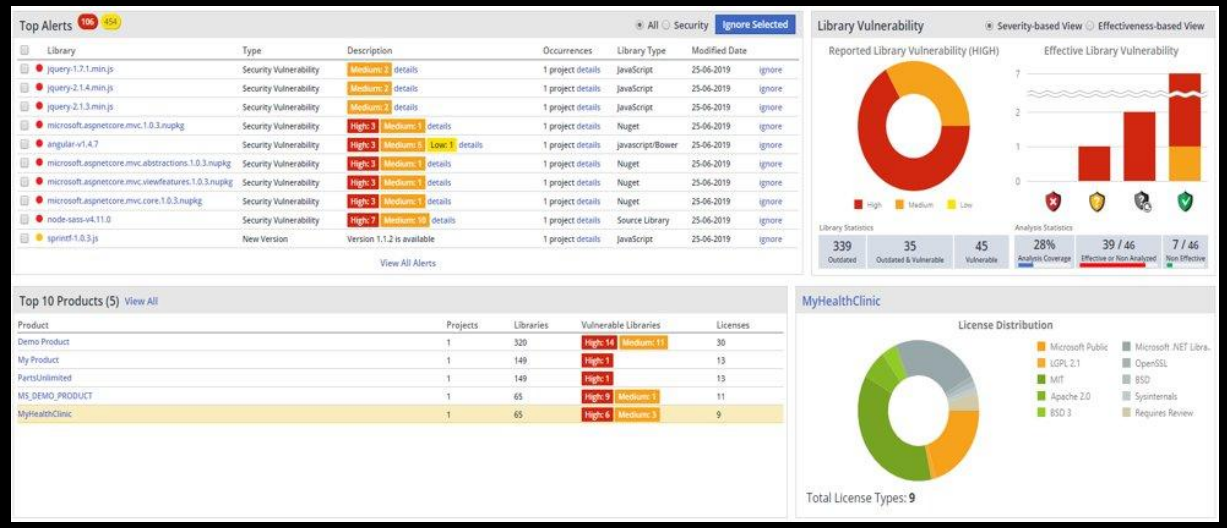
SCA - 治理

集中監控 OSS 元件
瞭解哪些專案使用了第三方元件
利用工具和流程輔助追蹤

瞭解開源軟體的許可風險

- Apache 2.0
- 跟
- BSD 3
- LGPL 2.1
- GPL 3.0
- Microsoft Public
- 公有土地

確保 OSS 安全
應對安全漏洞



SCA —— 為什麼它如此重要?

貴公司對所有發佈的產品負責。

攻擊者並不在乎誰編寫了存在漏洞的軟體。

當您必須通知客戶發生數據洩露時，他們也不會在意。

謝謝!