DevSecOps 和自動化安全檢測的敏捷導入

雲力橘子

2025/04/17

敏捷開發中~~整合安全檢測~~，並展示安全和效率可以並行不悖。

聽眾將學到~~實際操作方法~~，並理解如何改變團隊對於資安檢測的傳統觀點

About Me

Experience

- 雲力橘子 - 技術開發二部 副理
- 果核數位 - 軟體開發安全部 副理
- 資誠聯合會計師事務所 - 風險管理 顧問

Expertise

- 軟體安全開發與架構設計
- 流程改善與系統調教
- 資安評估與檢測

Certificate

- CISSP - Certified Information Systems Security Professional
- CSSLP - Certified Secure Software Lifecycle Professional
- ISO 27001:2022－ISMS Auditor/ Lead Auditor
- CEH - EC-Council Certified Ethical Hacker
- Java EE 5 Web Component Developer
- Solutions Developer Web Applications
- RHCVA - Red Hat Certified Virtualization Administrator
- Big Data Knowledge Today

Agenda

- 什麼是 DevSecOps 與敏捷
- 我們的實踐
- 重新思考敏捷的核心定位
- Take-away
-

什麼是DevSecOps與敏捷

敏捷宣言

gamania CloudForce 雲力橘子

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

個人與互動 > 流程與工具

可用的產品 > 詳盡的文件

與客戶合作 > 合約協商

回應變化 > 遵循計畫

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

DevOps 模型定義

DevOps 是集文化哲學、實務與工具於一身的結合，**可提升組織快速交付應用程式和服務的能力**：相較於使用傳統軟體開發與基礎設施管理程序的組織，這種作法能更快速地開發和改進產品。這樣的速度讓組織可以提供更好的服務給客戶，並在**市場上更有效率地競爭**。

DevOps 運作方式

在 DevOps 模型之下，開發與營運團隊不再「孤軍奮戰。」有時，這兩個團隊會合併成為一個團隊，讓工程師負責整個應用程式生命週期中的工作，包含從開發和測試、部署以及營運，並發展出許多不限於單一部門的技能。

Reference : https://aws.amazon.com/tw/devops/what-is-devops/

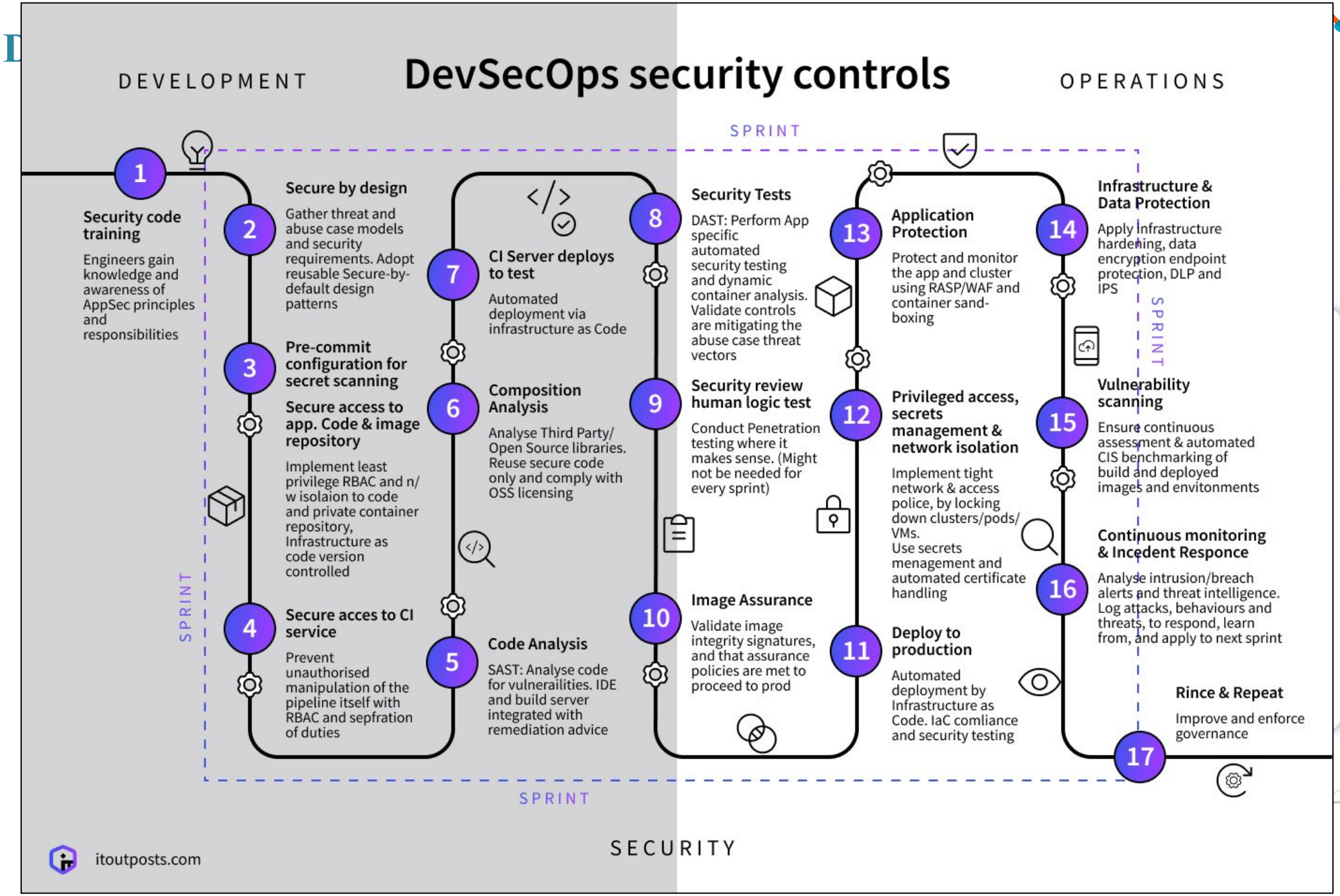
DevSecOps

在一些 DevOps 模型中，品質保證和安全團隊在整個應用程式生命週期，也能更緊密地與開發和營運團隊整合。DevOps 團隊中的每個人都專注於安全性時，有時稱為 DevSecOps。

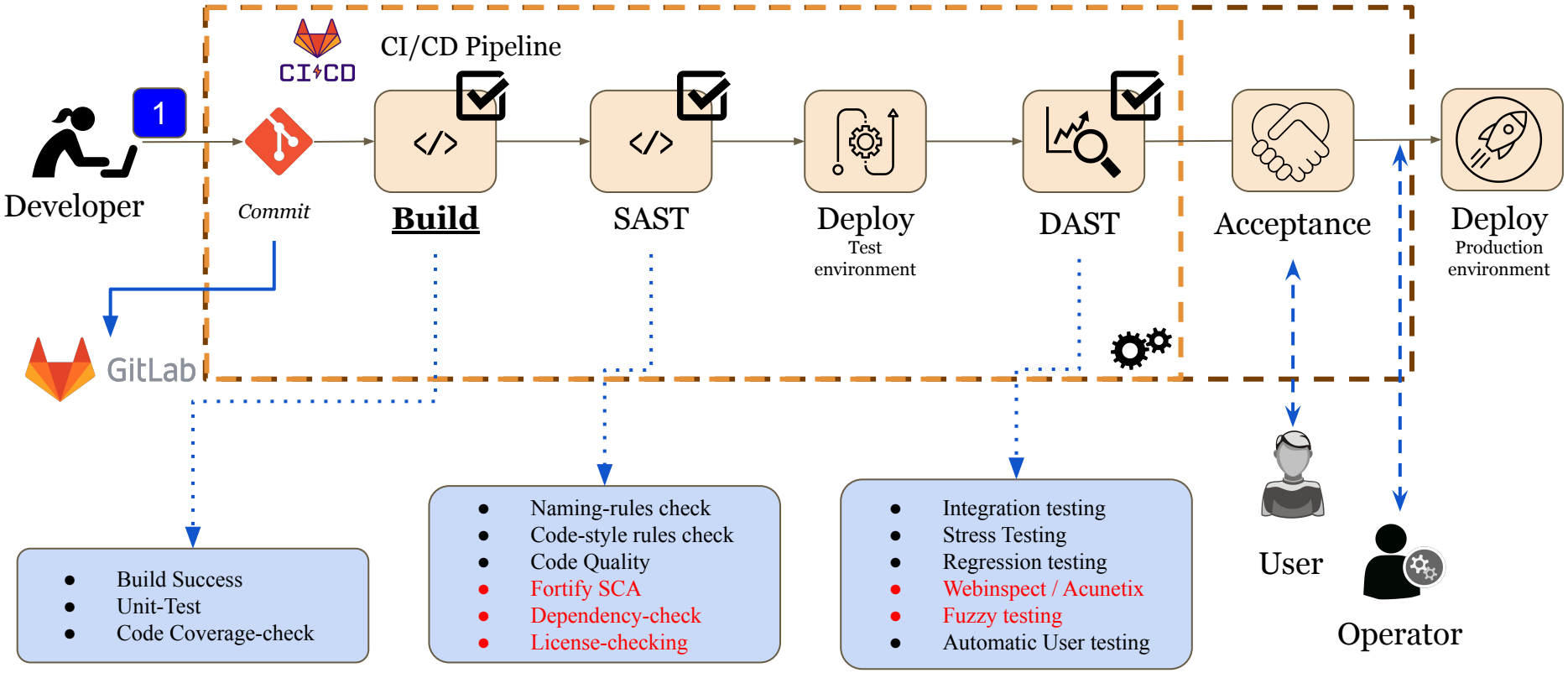
這些團隊會使用各種實務，**將以往手動且緩慢的程序自動化** 。他們使用可協助快速且可靠地營運與發展應用程式的技術堆疊與工具。這些工具也可協助工程師獨立完成通常需要其他團隊協助的工作 (例如部署程式碼或佈建基礎設施)，而這樣可以進一步**提高團隊的工作速度** 。

Reference : https://aws.amazon.com/tw/devops/what-is-devops/

敏捷與 DevSecOps 常見誤區 - 品質與自動化


DEVELOPMENT
DevSecOps security controls
OPERATIONS
SPRINT
1
Security code training
Engineers gain knowledge and awareness of AppSec principles and responsibilities
2
Secure by design
Gather threat and abuse case models and security requirements. Adopt reusable Secure-by-default design patterns
3
Pre-commit configuration for secret scanning
Secure access to app. Code & image repository
Implement least privilege RBAC and n/w isolaion to code and private container repository, Infrastructure as code version controlled
4
Secure acces to CI service
Prevent unauthorised manipulation of the pipeline itself with RBAC and sepfration of duties
5
Code Analysis
SAST: Analyse code for vulnerailities. IDE and build server integrated with remediation advice
6
Composition Analysis
Analyse Third Party/ Open Source libraries. Reuse secure code only and comply with OSS licensing
7
CI Server deploys to test
Automated deployment via infrastructure as Code
8
Security Tests
DAST: Perform App specific automated security testing and dynamic container analysis. Validate controls are mitigating the abuse case threat vectors
9
Security review human logic test
Conduct Penetration testing where it makes sense. (Might not be needed for every sprint)
10
Image Assurance
Validate image integrity signatures, and that assurance policies are met to proceed to prod
11
Deploy to production
Automated deployment by Infrastructure as Code. IaC comliance and security testing
12
Privileged access, secrets management & network isolation
Implement tight network & access police, by locking down clusters/pods/ VMs.
Use secrets menagement and automated certificate handling
13
Application Protection
Protect and monitor the app and cluster using RASP/WAF and container sand-boxing
14
Infrastructure & Data Protection
Apply infrastructure hardening, data encryption endpoint protection, DLP and IPS
15
Vulnerability scanning
Ensure continuous assessment & automated CIS benchmarking of build and deployed images and envitonments
16
Continuous monitoring & Incedent Responce
Analyse intrusion/breach alerts and threat intelligence. Log attacks, behaviours and threats, to respond, learn from, and apply to next sprint
17
Rince & Repeat
Improve and enforce governance
SPRINT
SPRINT
SPRINT
SECURITY
itoutposts.com

Try with Automation



Automation Test Evolution

gamania CloudForce 雲力橘子

JUnit

BenchmarkDotNet
Powerful .NET library for benchmarking

POSTMAN

Open API Specification
Swagger
3.0

shutterstock.com · 2133924155

#hidden 奇異幻想的安全測試剛剛好

Gitlab 測試結果呈現

gamania CloudForce 雲力橘子

Merge branch 'feature/[redacted] 'develop'

Delete

Passed **Lung** created pipeline for commit `4d71d9f7` 1 day ago, finished 1 day ago

For `develop`

11 jobs 22 minutes 22 seconds, queued for 1 seconds

Pipeline Jobs 11 **Tests** 80

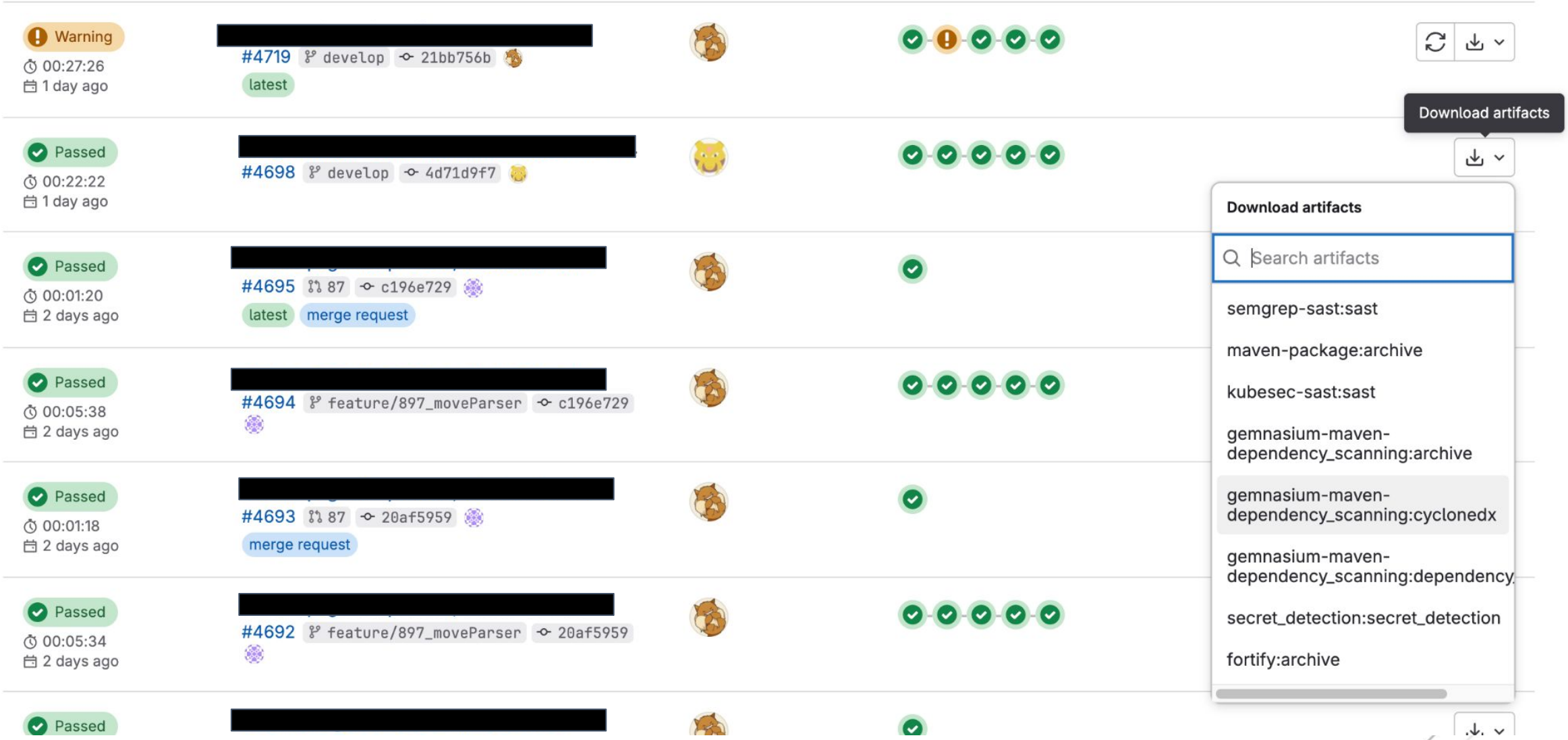
Summary

80 tests | 0 failures | 0 errors | 100% success rate | 23.40s

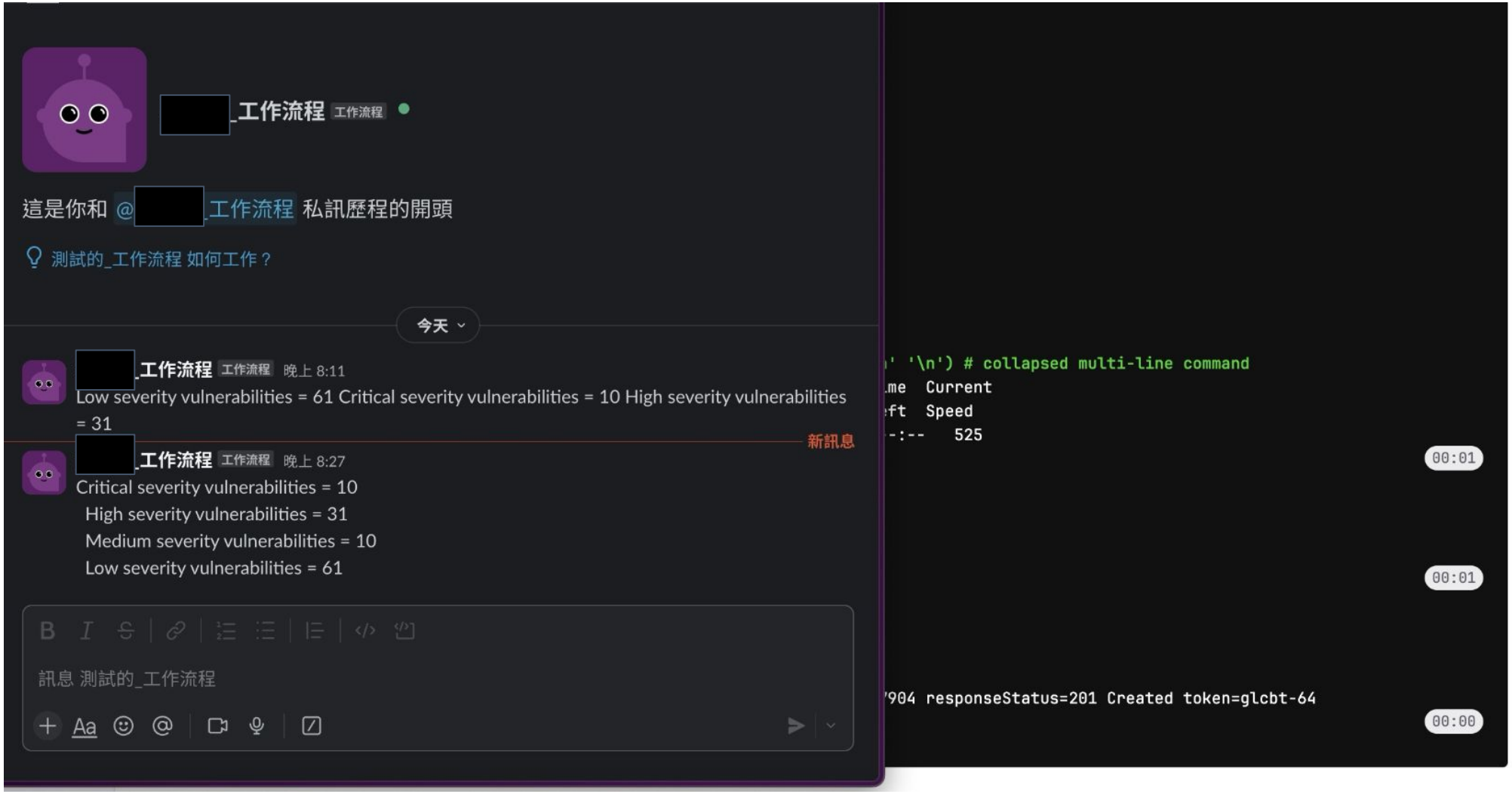
Jobs

Job	Duration	Failed	Errors	Skipped	Passed	Total
maven-unit-test	23.40s	0	0	0	80	80

Gitlab Pipeline Artifact 管理執行結果



自動化通知流程



Queued: 2 seconds
Timeout: 1h (from project)
Runner: #19 (7szNcdWi) workstation
Tags: docker

Job artifacts
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep | Download | Browse

Commit `1e9ff5c9`
在報告生成步驟中新增中等和低嚴重性漏洞計數，並更新摘要報告內容以包含所有漏洞級別

Pipeline #3298 Warning for `feature/ci_sca_risk_count`

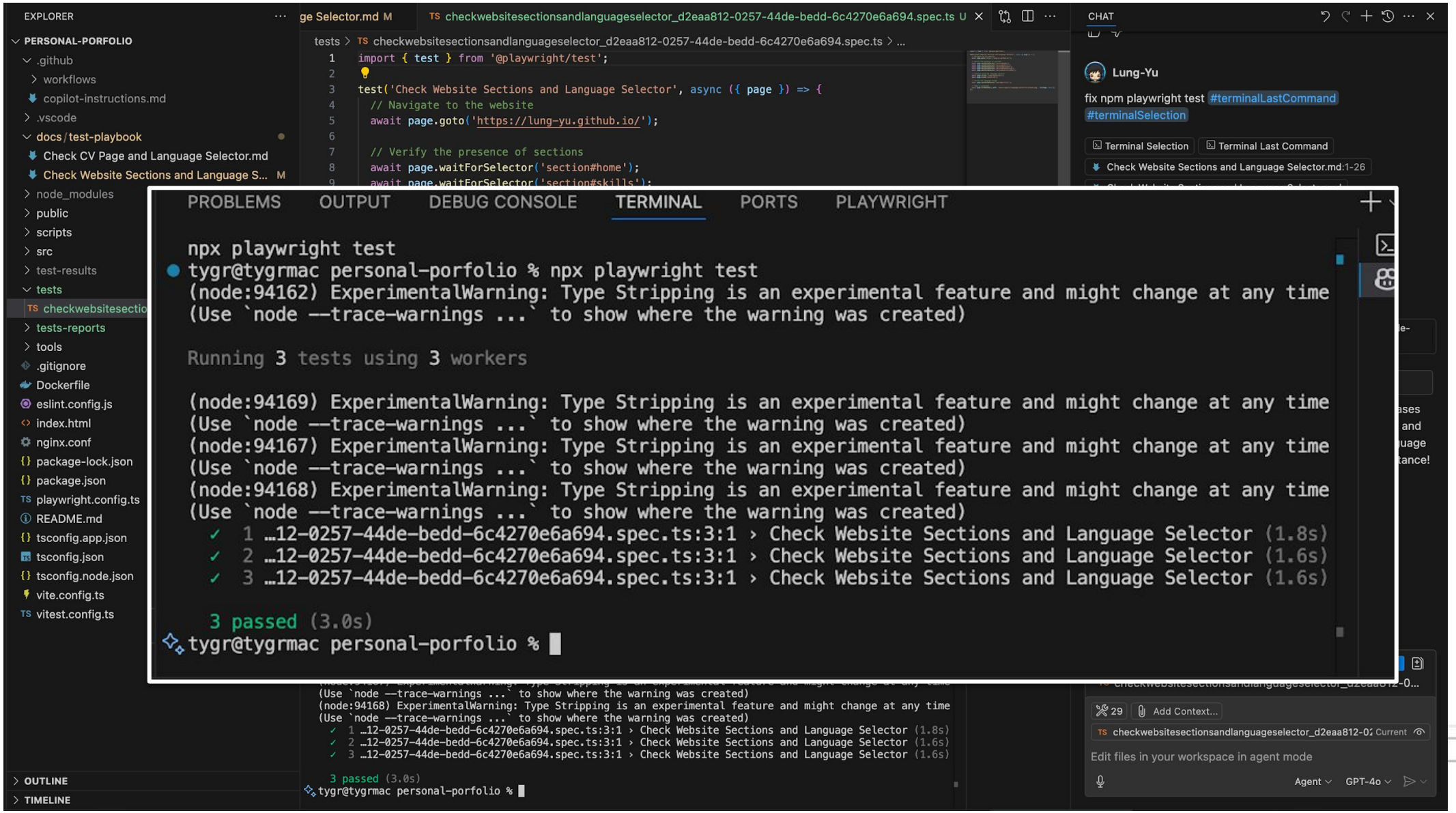
reports

Related jobs
→ parse_fortify_report

搭配 AI +Playwright MCP 實踐測試自動化

gamania CloudForce 雲力橘子

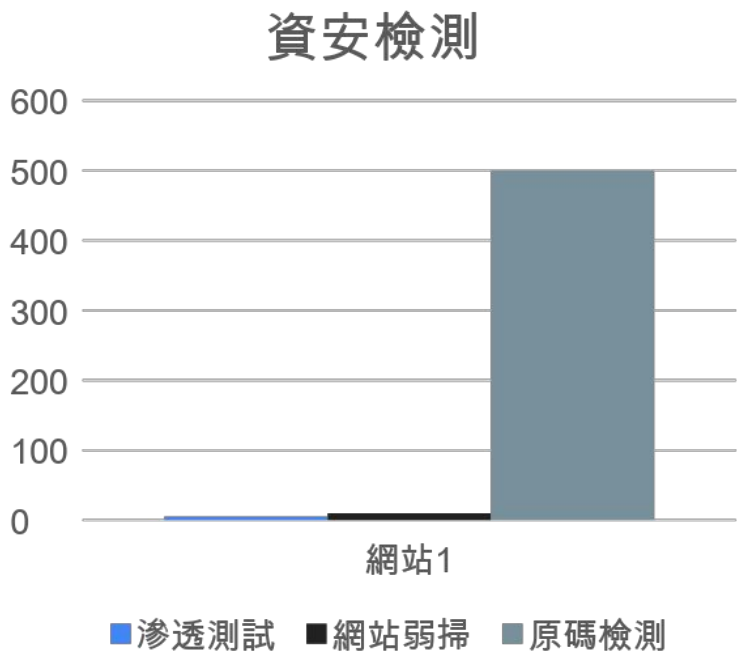
#hidden 奇異幻想的安全測試剛剛好



敏捷與DevSecOps 常見誤區 - 安全檢測

自動化安全檢測

- Penetration Test
- Website Scanning
- Source code Scanning
- Fuzzing Test
- OS Vulnerability Scanning



當時的經驗總結

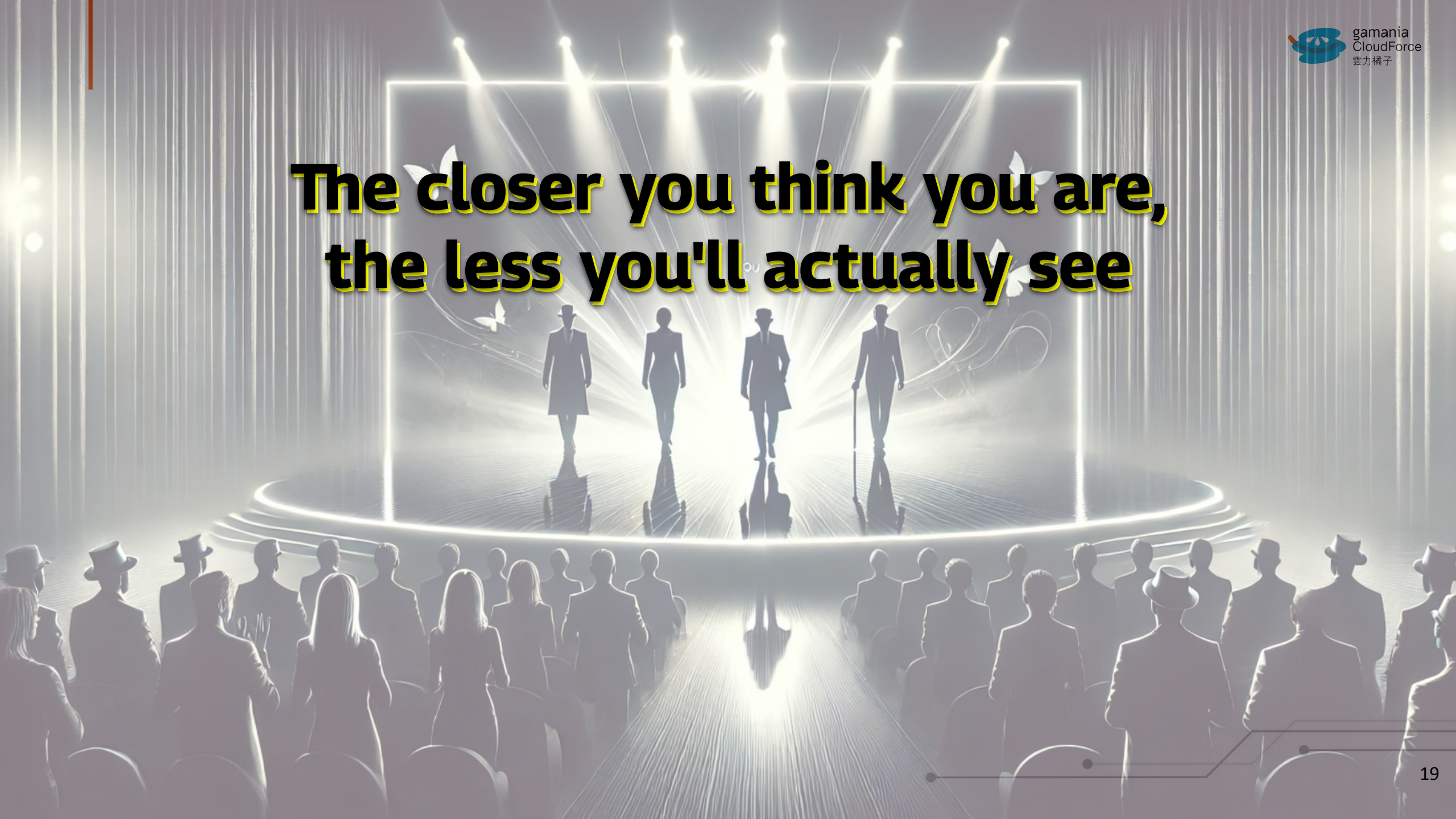
- 企業的威脅邊界
- 使用共通認可之檢核套件
- 建立企業審核通過之程式碼片段

// 全員都有資安認證

// 架設主機與系統安全維運

// 系統開發與產品開發

DevOps + Sec 是強力援手還是搗亂者

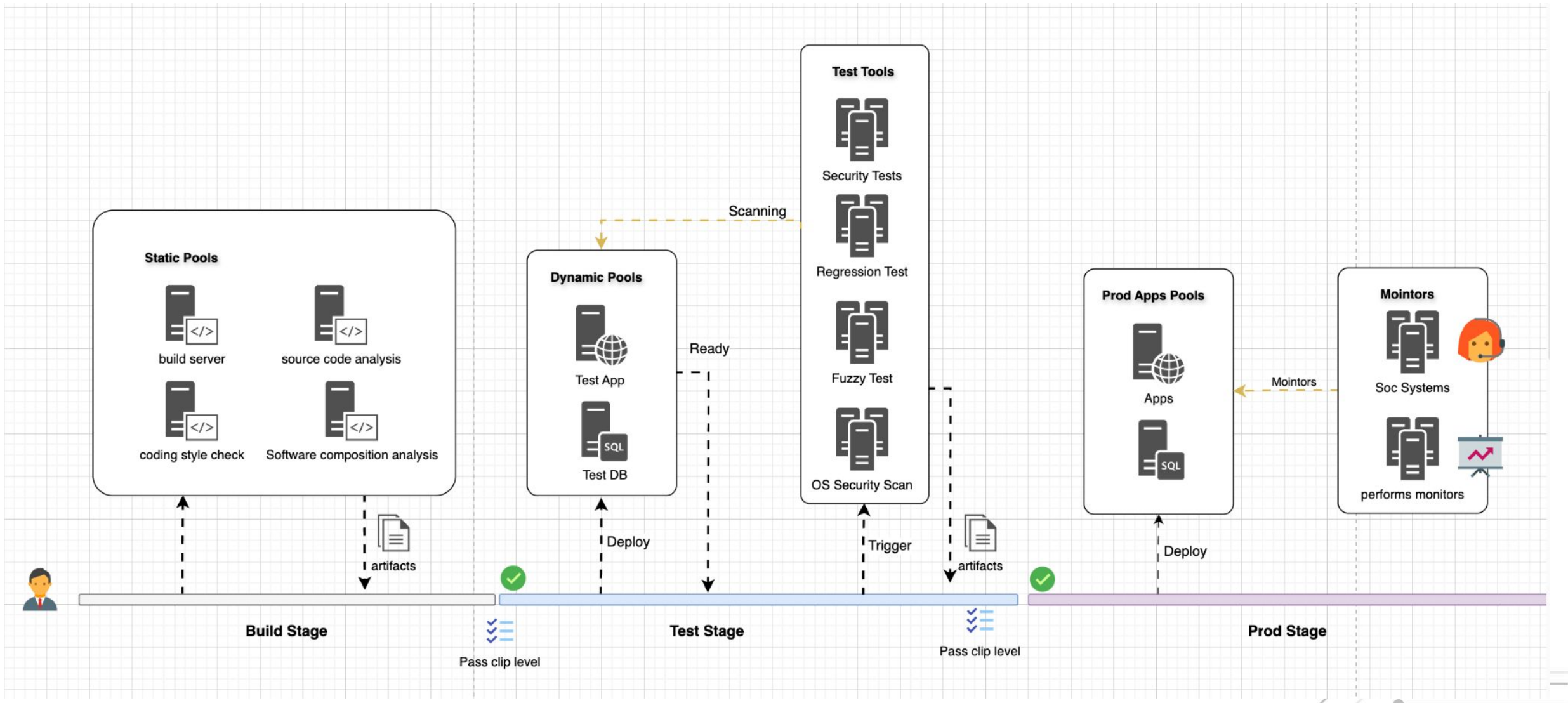

The closer you think you are,
the less you'll actually see

重新思考敏捷的核心定位

透過 Scrums 循環精進團隊

- 發現問題
- 找到根因
- 解決問題

開發建置流程與管線(Pipeline)



Retro 復盤與分析困境

1. 開發人員在諮詢顧問人員的時候耗時等待影響上版
2. 資安與開發技能取向差異，人員難以全數掌握
3. 資安顧問人力遠低於開發人力

Retro 檢討並進行改進

AI 資安顧問 24hr 提供咨詢服務

4/17 (四) 10:15 - 10:45　7F 701F　Secure Software & DevSecOps 論壇

讓修補經驗成為懂你系統的資安顧問

在資安掃描與修復的過程中，我們常面臨一個關鍵挑戰：如何將通用的資安建議轉化為符合系統實際情況的修補方案。將分享如何將累積的弱點修補經驗，結合 RAG 技術，打造一個真正理解系統的智慧資安顧問。

從實戰經驗出發，我們將深入探討如何建立完整的知識庫，包含各種弱點的修補策略。透過整合這些知識，打造出能夠提供...

詳細內容 →

講者

林秉正

雲力橘子

技術開發二部 主任

Retro 復盤與分析困境

gamania CloudForce 雲力橘子

1. AI 顧問是針對已經發現的問題進行分析與應對
2. 資安左移，開發人員在一開始就能避免？
3. 已經有威脅建模、資安專家協同討論、安全程式範例與套件

必要時一起討論與分析

Mass Assignment: Insecure Binder Configuration

弱點說明:

為了簡化開發並提高生產力，大多數的現代架構都允許自動個體化物件，並填入名稱與要繫結之類別的屬性相符的 HTTP 要求參數。自動個體化並填入物件可以加速開發，但是如果實作不慎，就可能會導致嚴重問題。已繫結類別或巢狀類別中的任何屬性將會自動繫結至 HTTP 要求參數。因此，惡意的使用者便能夠將值指派給已繫結或巢狀類別中的任何屬性，即使這些類別並未透過網頁表單或 API 約定而暴露給用戶端。

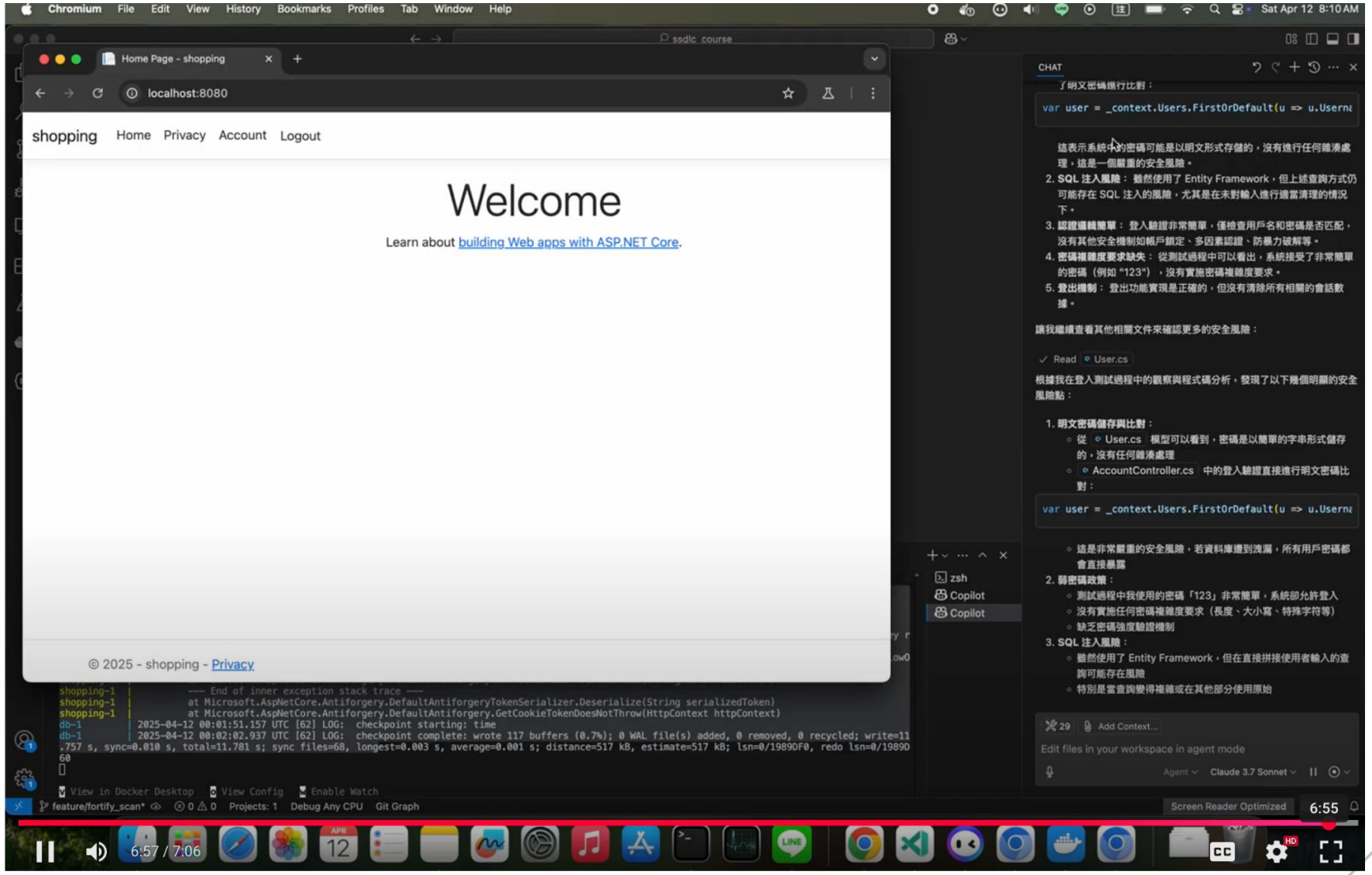
修補方法:

遵循「最小權限原則」針對綁定(Bind)的物件(Class)進行限制妥善的限制，並進行明確的聲明，如:DataMember、DataContract、Required、BindRequired。限制語法文獻可參考MSDN(https://learn.microsoft.com/zh-tw/dotnet/framework/wcf/feature-details/specifying-data-transfer-in-service-contracts)

並且於HTTP-Method參數設定[Bind(Include = "Number, Combination")]相關限制。

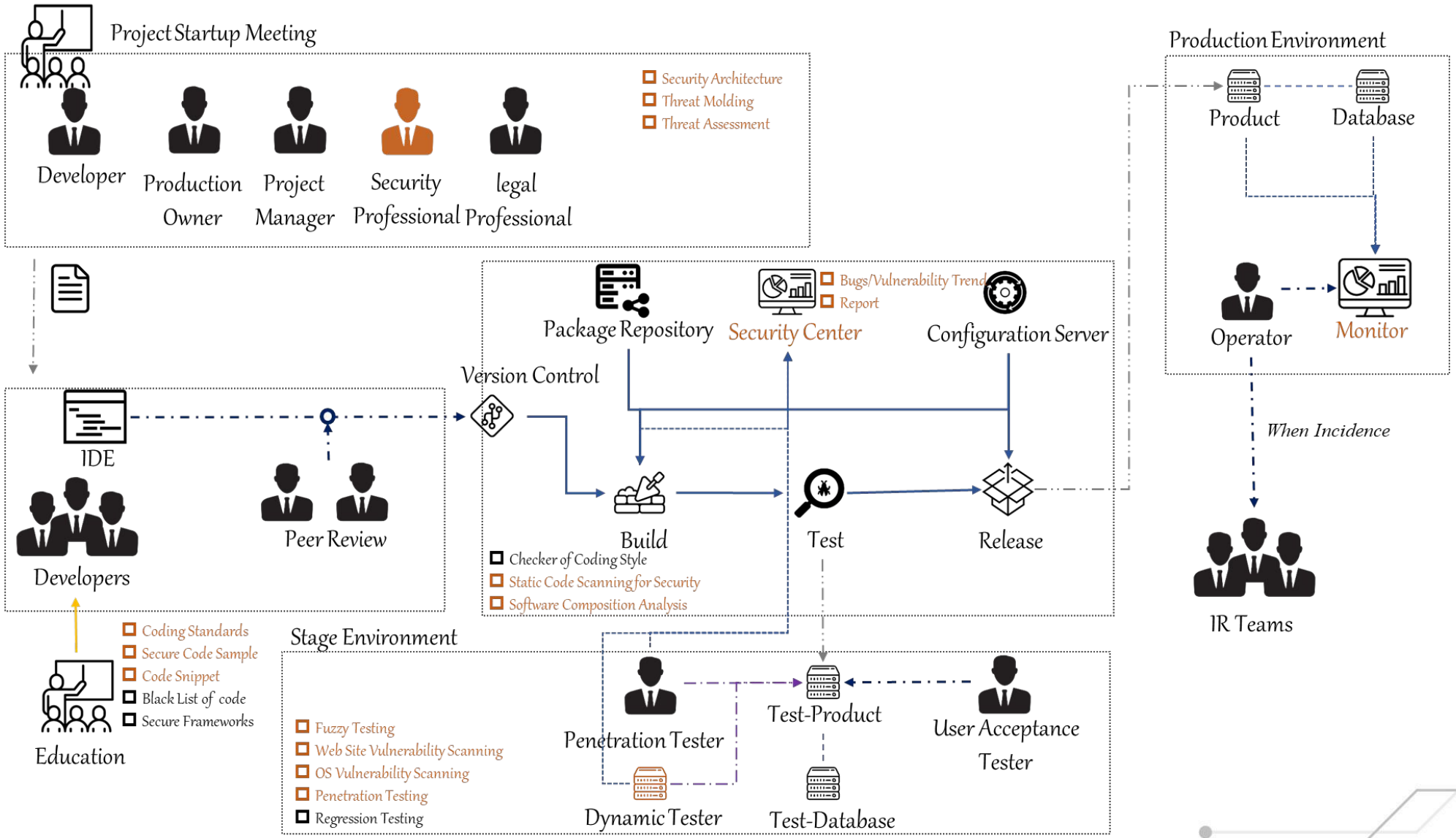
```
[HttpPost]
public ActionResult Add([Bind(Include = "name,Title,Combinations")] Student student)
{
...
}
```

AI 搭配 MCP 進行程式安全性分析



#hidden 奇異幻想的安全測試剛剛好

現階段的流程做法可能不完美，但他還會持續進化



Summary

- 自動化與AI的深度整合
  - 在敏捷流程中引入自動化安全檢測，提升檢測效率。
  - 利用AI技術輔助資安檢測，減少人力不足的困境。
  - 讓技術演進成為團隊實踐的核心推動力。
- 資安左移與合作精神
  - 強調資安檢測在開發初期的重要性，避免事後補救。
  - DevSecOps重點在於合作而非分工，促進團隊的協同效應。
  - 每次Scrum迭代都是提升安全性的契機。
- 持續反思與流程優化
  - 透過Retro復盤分析，識別並解決流程中的困境。
  - 不斷重新思考敏捷的核心定位，以適應快速變化的需求。
  - 以精益求精的態度改進DevSecOps流程與架構。

What i want to says

gamania
CloudForce
雲力橘子

每個複雜的問題都有一個簡單
明瞭 且錯誤的答案

As Sec & Dev

gamania
CloudForce
雲力橘子

Thanks for listening

ISC2 CHAPTER | TAIPEI

gamania CloudForce 雲力橘子

【關於我們】

ISC2 Taipei Chapter 是 ISC2 在台灣的分會組織，致力於推廣資安專業知識、交流與社群互助。成員包括經 ISC2 認證的資安專家，及各領域資安專業人士。

【成立宗旨】

提升台灣資安專業社群的能見度與凝聚力。

【使命與活動】

- 推廣資安認證與專業發展
- 舉辦技術講座、各種資安研討活動
- 促進會員交流與跨界合作
- 參與國際資安社群互動

https://www.isc2chapter.tw

contact@isc2chapter.tw

加入 ISC2 Taipei Chapter 讓您的資安之旅，有專家同行！