前言

雲端安全英雄們，大家好！你是否已經厭倦在成堆的安全工具之間疲於奔命，像小狗追著尾巴跑一樣追查警報，最終淹沒在一望無際的通知訊息汪洋之中？別煩惱了，因為這本書將徹底翻轉我們守護雲端王國的做法。

這不是你阿公的那本布滿灰塵的安全手冊，我們將打破各自為政的心態，解開資訊碎片化的枷鎖。本書就是你的 CNAPP 百科全書。

繫好安全帶，加入我們。本書將帶你進入一個處處都有精實的安全機制、團結的團隊，且威脅無所遁形的世界。讓我們一起征服雲端，將它打造成抵禦網路惡棍的堡壘！

誰應該閱讀這本書

安全是所有人的責任，本書適用於涵蓋 DevOps 和 SecOps 的各種技術領域。不過，最重要的是，本書是為那些勇於挑戰混沌不明的雲端原生應用程式開發領域的人而寫的。本書假設讀者已具備雲端原生生態系統的基本知識。

為什麼我們要寫這本書

雲端原生應用程式安全領域的縮寫多如牛毛，幾乎到了讓人無法有效溝通的臨界點。當研究顧問公司 Gartner 提出「CNAPP（Cloud Native Application Protection Platform）」這個縮寫時，解方隱然成形，這是一統天下的縮寫啊！然而，我們的產業卻爭先恐後地

定義 CNAPP，這反而帶來更多混亂。在這本書中，我們希望釐清雲端原生安全的平台解決方案有什麼技術面的利益，並且探討它為文化帶來的優勢。

如何閱讀這本書

為了讓 CNAPP 的概念更加具體，並分享真實世界的安全情境，我們用兩個主軸來貫穿這本書：

1. 有一通來自 MI5 的電話動搖了某個組織的應變流程根基。
2. Log4Shell…聽過嗎？我們將運用這個真實的情境，透過一系列的情節來展示 CNAPP 的威力，並深入探討防禦概念，包括預防、偵測，以及回應當今最複雜的威脅。

以下是各章的重點。

第 1 章將鋪陳背景，說明攻擊者與防禦者的對峙情勢，深入探討攻擊面擴大、團隊與資訊孤島化，以及警告訊息超載的問題，並引入貫穿本書的故事，最後回到安全的首要原則。

第 2 章介紹可觀察性，並轉化它，在現有的安全孤島中，實現一致的安全政策。

第 3 章從 CNAPP 的誕生地正式開始我們的技術旅程，我們要探索雲端安全態勢（cloud security posture）。

第 4 章將討論焦點左移到工具、文化與協作層面。

第 5 章與第 6 章探討供應鏈安全問題，從應用程式內的直接與間接依賴元件談起，再進一步探究流程（pipeline）本身的安全性。

第 7 章揭露如何將執行環境的雲端安全工具整合為一股強大力量。你的雲端安全態勢、身分識別機制、作業單元保護措施將齊心合作，將海量的警報轉變成可行動的見解。

第 8 章要問的是：「你的資料在哪裡？」，資料對攻擊者而言是位於偉大航道終點的大秘寶。我們必須自問：它在哪裡？攻擊路徑是否正好指向那裡？

第 9 章要向部落知識說不，讓你的安全夢幻團隊擁有前所未有的協作知識。雲端原生創作者、建構者與防禦者終於開始攜手合作，成為一面堅不可破的護盾。

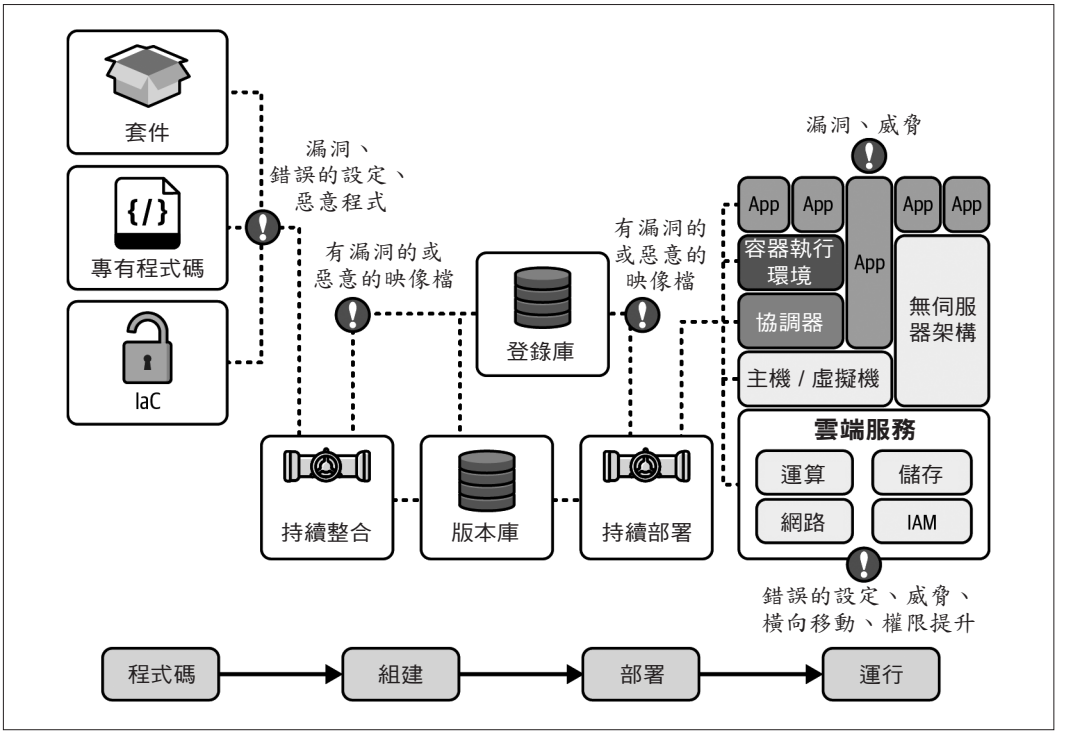

圖 1-3　在整個雲端原生軟體開發生命週期中可能遭受攻擊的各個環節

打從撰寫程式碼的階段開始（如今也包括透過 IaC 工具來撰寫基礎架構），專有程式碼、套件，甚至開源工具都可能引入漏洞。你也會有錯誤的設定，甚至被植入惡意程式。

隨著生命週期進入持續整合階段，有漏洞的惡意映像檔可能會在建構容器映像檔的階段生成。然後，這些映像檔會經由持續部署流程進入正式環境，並經由叢集協調器，在應用、叢集、伺服器、容器、無伺服器函式等軟體層之中實體化，它們都在受託的虛擬機器與底層的雲端基礎服務之上運行，雲端基礎服務包括運算、儲存、網路，以及身分與存取管理。如此快速的變更流程，可能會因為錯誤的設定或執行環境之中的惡意程式，在正式環境中進一步造成漏洞。

Act（行動）

透過整合的視角，在每一個領域採取正確的行動，主動與被動地保護系統免受威脅或漏洞的影響。

在電影《捍衛戰士（*Top Gun*）》裡行之有效的策略，同樣適用於你和你的安全團隊。你的協作做得越好，就會有越多正向的行為「湧現」出來，OODA 迴路就會越完善，最終，你與團隊就能更快速學習並適應新的安全挑戰。

CNAPP 可以啟動你的雲端原生安全 OODA 迴路

在面臨雲端原生環境的複雜性、規模與變化速度時，建立、支援與促進整個團隊的安全 OODA 迴路正是 CNAPP 應該發揮的核心價值。來自不同廠商的功能或許有所差異，但你想從 CNAPP 獲得的幫助是不變的，也就是讓你的開發、安全工程，與安全營運團隊能夠有效地合作，如此一來，你就能夠在整個軟體開發與交付生命週期中，建立快速的安全 OODA 迴路，如圖 1-8 所示。

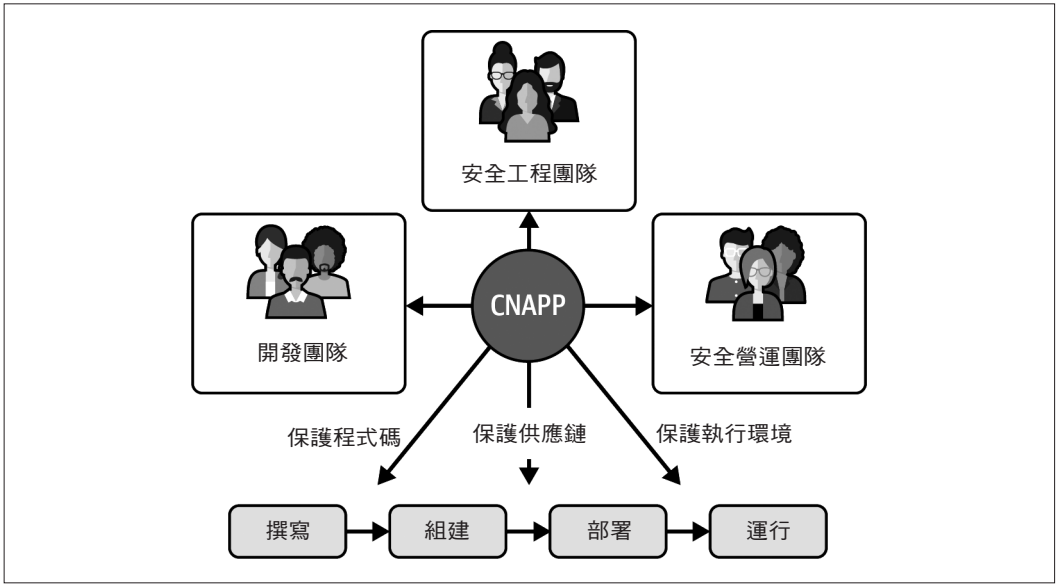

圖 1-8　你的 CNAPP 會將所有團隊與整個雲端原生棋局連接起來

CNAPP 的整合功能扮演雙向結締組織（connective tissue）的角色，將 CNAPP 掌握的安全態勢和進行中的漏洞修復行動連接起來。不過，珍貴的回饋迴路不是只有這一個而已，所謂的修復行動不是只有跟著指示，把該修的地方修一修而已。你將發現，就像我們的經驗，你可以做的事情、可以改善之處、可以學習的事情還有很多。

學習：新問題、新政策、新控管 —— 新的迴路

有一位智者說過：「後見之明是一種強效的迷幻劑」。[16] 我想補充：「以為自己可以完美預測未來，也是不遑多讓的迷幻劑」。事後的見解與事前的自信都會讓人陷入「一切都很穩定、也都在我們的掌握中」的假象，讓人們誤以為安全威脅是可以輕易預測的 —— 「我們已經面面俱到了，安全得很！」。然而，它們也會讓你忽略不符合既定印象的攻擊模式 —— 你會漏掉那些看起來「太瞎」而不可能出現的手法。

若要從 CNAPP 揭露的安全資訊中學到東西，你就必須拋棄這兩張虛假的安全網。在 CNAPP 裡，現成的安全威脅與法遵框架看起來非常完整，或感覺起來很完整，這很容易讓你落入「一切都在掌握中」的錯覺，甚至因此而過度自信，以為自己有能力應付所有狀況，這其實非常危險。事實上，CNAPP 對於安全與法遵的評估永遠沒有「完成」的一天。CNAPP 不僅會從外部的資訊來源中持續學習，也會從你自己的發現中學習，藉機讓它在你的環境中更有效。你可以打造出你自己的專屬 CNAPP。

你的 CNAPP 是可自訂的。預設的安全威脅與法遵框架的資料來源與輸入只是起點，當你深入探索自己的系統與業務領域時，你可能會發現更細膩的 CNAPP 安全政策、更微妙的威脅，還有更符合你需求的法遵檢查方式，它們都是可供學習的機會。

在雲端原生安全環境中，學習不是錦上添花（nice-to-have），而是關鍵所在。如第 1 章所述，學習能夠幫你適應這一個包含系統與團隊在內的動態複雜環境。CNAPP 的安全政策與全方位協作回饋迴路確實很棒，但學習可以再加上一個新迴路，讓你能夠不斷質疑、調整、優化 CNAPP 賴以運作的前提，如圖 3-13 所示。

16 這句朗朗上口的說法來自 John Allspaw（*https://oreil.ly/uYn0F*），當時他說的主要是「事後回顧」或「從結果倒推原因」往往會讓人錯過如果當初親身經歷事件的話，實際可以觀察並學到的重要教訓。

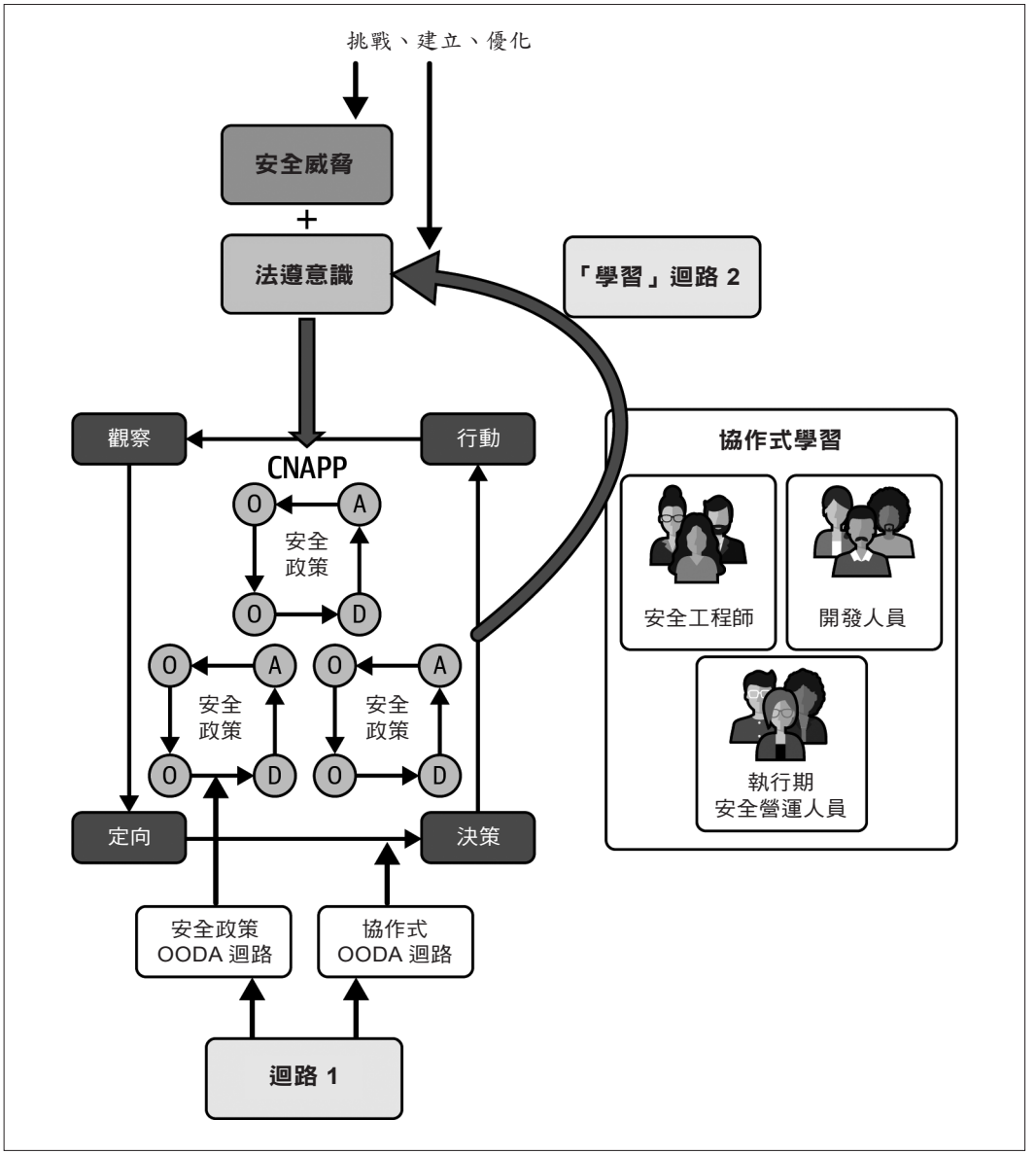

圖 3-13　加入「學習」迴路來為你的 CNAPP 回饋迴路補上最後一塊拼圖。透過學習，你可以質疑、調整、優化 CNAPP 據以運作的安全威脅與法遵框架

真實案例

雖然有很多案例可供選擇，但有兩個案例因為手法複雜且新穎而特別引人注目。這兩個案例正是上述的 OWASP 十大 CI/CD 安全風險（如圖 6-3 所示）得以誕生的靈感來源。

Codecov

在 2021 年 4 月，攻擊者利用 Codecov 軟體供應鏈之中的一個漏洞發動了攻擊。以下是這一起事件的來龍去脈：

初步入侵

這起入侵事件，始於攻擊者在未獲授權的情況下，存取了 Codecov 的基礎架構。雖然確切的入侵手法尚未公開，但極可能涉及釣魚、憑證填充攻擊（credential stuffing）（*https://oreil.ly/Togno*）或是利用 Codecov 系統的漏洞。

依賴元件混淆攻擊 *CICD-SEC-3*

進入 Codecov 的基礎架構之後，攻擊者執行了一種稱為「依賴項目鏈濫用（dependency chain abuse）」的供應鏈攻擊。他們發現 Codecov 的組建伺服器被設定為從 PyPI（Python Package Index）、npm（Node.js 套件管理器）等公用套件庫自動抓取與安裝軟體。接著，攻擊者製作了開源工具的惡意版本，例如由 Codecov 提供、被廣泛使用的「bash uploader」腳本，該腳本原本是用來上傳覆蓋率報告的。他們將這些惡意版本上傳至公開的版本庫，模仿合法套件的名稱，但裡面的程式碼已被修改，加入後門。當使用者執行被植入後門的 bash uploader 腳本來上傳覆蓋率報告時，惡意程式碼便會執行，讓攻擊者能夠從整合了 Codecov 的環境中外洩敏感資訊。

資料外洩

已被攻擊者竄改的惡意腳本包含一些指令，它們會蒐集並傳送環境變數（其中包含敏感的憑證與權杖）到攻擊者所控制的外部伺服器，讓他們能從被入侵的環境中，讀取並外洩敏感資料，可能包含原始碼、憑證，以及其他機密資訊。

影響與後果

這起安全事件造成的影響非常深遠，可能波及數千個組織及其軟體專案。雖然 Codecov 迅速採取補救措施來減輕這起事件的影響，但受害者仍面臨原始碼遭受未授權存取、敏感憑證外洩、甚至系統被進一步入侵……等風險。

SolarWinds

SolarWinds 的安全事件是在 2020 年 12 月被發現的，它是近年來最複雜的網路攻擊之一，涉及 SolarWinds 的 Orion IT 管理軟體被入侵。以下深入解析這一個複雜的攻擊手法：

初步入侵

攻擊者可能是由某一個國家支持的組織，最初，他們入侵了 SolarWinds 的組建環境或軟體供應鏈。他們取得 SolarWinds 軟體開發基礎架構的存取權，手法可能是釣魚攻擊、密碼噴灑攻擊（password spraying）（*https://oreil.ly/P0utV*），或是利用 SolarWinds 所使用的第三方軟體之中的漏洞。

攻擊者在 SolarWinds 的開發環境中，將後門植入 SolarWinds Orion 平台的原始碼之中。這個精心設計的後門能夠與合法程式碼融為一體並躲避偵測，讓傳統的安全防護機制難以察覺，如圖 6-4 所示。這可以直接對應 OWASP CI/CD 十大風險（CICD-SEC-9：未能妥善管理第三方服務的使用（*https://oreil.ly/-iCBm*））。

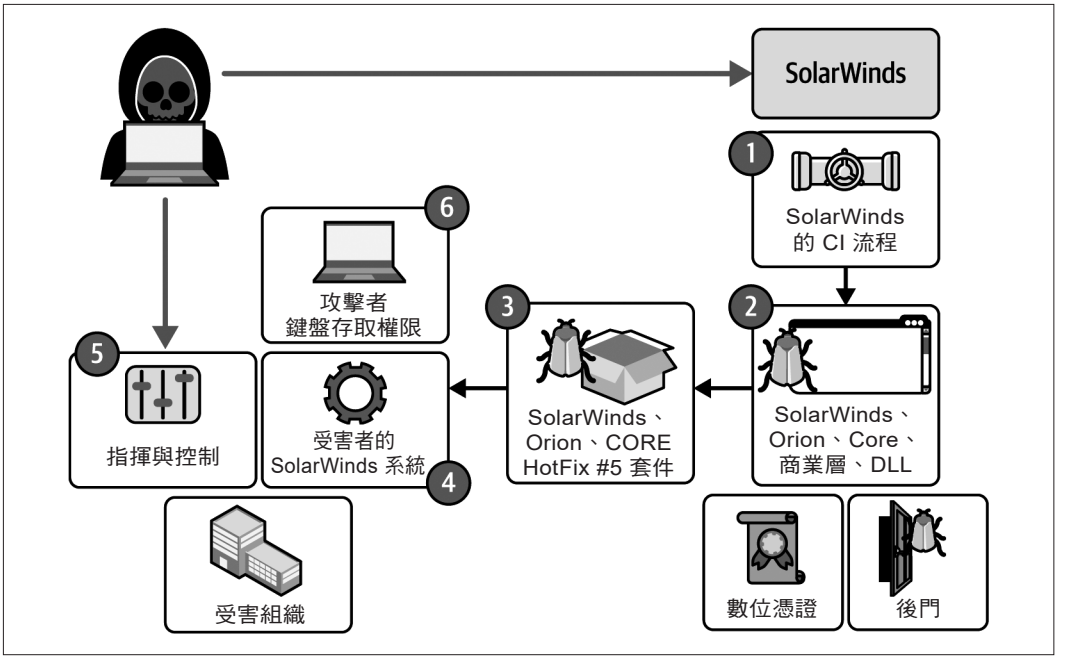

圖 6-4　攻擊 SolarWinds Orion 的路徑