第一章

零信任的基礎概念

網路監控無處不在，到底還有誰可以信任，將光纖拉到你家的網路供應商？或是替資料中心佈線的外包商？真的相信網際網路流量不會被竊聽嗎？當然不相信！

Edward Snowden（斯諾登）和 Mark Klein 等深喉嚨揭開政府支助的綿密間諜網，費盡心思成功切入大型集團的資料中心，此事件震驚了全世界。為什麼呢？當擁有和他們一樣的機會時，尤其知道網路流量都不加密，是否也想這麼幹？

不要認為資料中心內部的系統和網路流量可以被信任，這種假設本身就是漏洞。現今的網路形態和使用模式，已不是邊界防禦思維可以應付的，當駭客入侵某一部主機或網路後，要在「安全」架構內部自由行動，根本輕而易舉。

零信任的目標是要解決人們在網路活動中長久以來的信任問題，儘可能達成有效保護網路通訊及授權存取，不必刻意理會傳輸層的安全機制。當然，這是個遠大的目標，好消息是現今的加密技術已不可同日而語，再加上強而有力的自動化系統，此願景確實可以成真。

關於零信任網路

零信任網路的概念是基於五項假定：

- 網路始終受到對手覬覦。
- 網路總是存在外部和內部威脅。
- 上網位置不能作為決定信任度的依據。

- 對所有設備、使用者和網路流量都須通過身分驗證，並取得存取授權。
- 安全政策必須可動態調整，並盡可能以各種資料來源做為評估依據。

傳統的網路安全架構會部署一套或多套防火牆，將網路（或某個網段）劃分成不同區域，每個區域被授予某種程度的信任，藉以決定存取網路資源的權限。此模型宣示濃厚的縱深防禦，例如，面向網際網路的 Web 伺服器被視為具高度風險的資源，應部署在可嚴格監控和控制流量的緩衝區（通常稱為 DMZ；非軍事區）。這種作法產生一種讀者之前可能看過的類似架構，就如圖 1-1 所示。

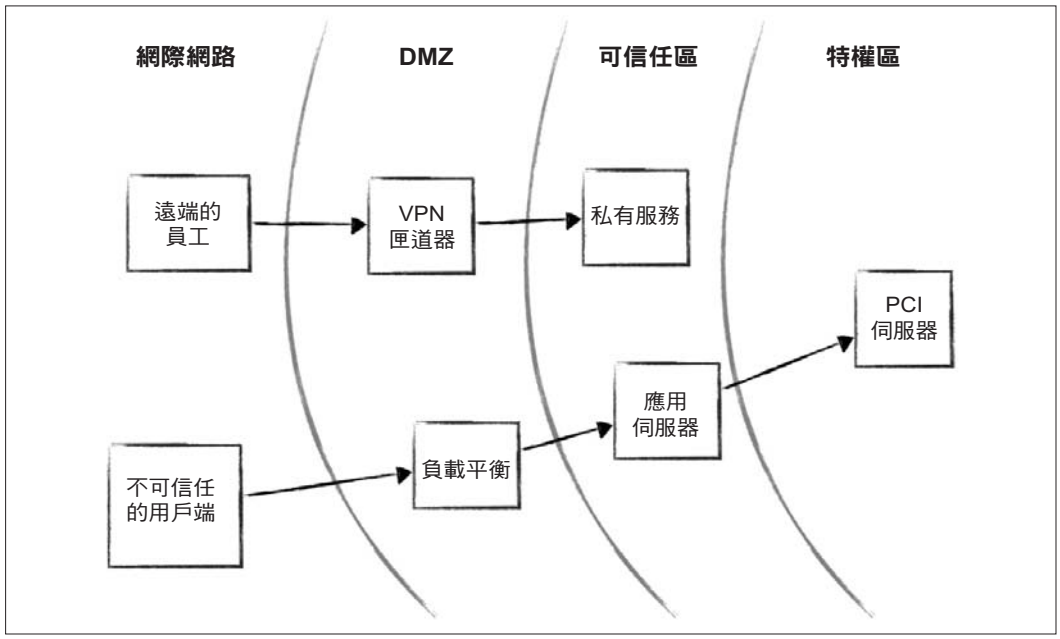

圖 1-1　傳統的網路安全架構

零信任模型則完全翻轉這種架構，在昔日看來堅實的措施，如今不過是一時權宜，面對現代的網路攻擊型態，已然捉襟見肘、力有未逮。傳統架構主要缺點如下：

- 缺乏內部區域的流量檢查。
- 主機的實體或邏輯部署都缺乏彈性。
- 存在**單點失效**（SPOF）全軍覆沒的缺陷。

現在，假設對手僱用一批駭客，想要取得貴公司的庫存清冊和銷售數量。駭客從網際網路收集貴公司員工的電子郵件位址，並發送電子郵件給這些員工，內容是假造辦公室附近的某家餐廳正舉辦優惠活動。果不其然，某位員工點擊了郵件上的鏈接，讓攻擊者能夠在他的電腦上安裝惡意軟體，該惡意軟體透過背景連線通訊，在攻擊者與受駭電腦建立連線。還好，這位員工只是一名實習生，存取權限層級並不高。

駭客開始搜尋網路，發現貴公司在網路上使用某套檔案分享軟體，網路上的所有員工的電腦都沒有升級到最新版本，現行版本存在最近所公佈的漏洞，駭客地毯式搜尋具有較高存取權限的電腦（攻擊者若具備高階攻擊知識，過程會更有針對性），最終找到網頁應用程式設計師的電腦，駭客在這台電腦上安裝鍵盤側錄程式，以便找出登入 Web 伺服器的身分憑據（帳號及密碼），然後利用收集到的憑據，透過 SSH 連入 Web 伺服器，並使用設計師的 sudo 權限從磁碟中取得資料庫密碼，接著連接到資料庫。現在，駭客轉存並下載資料庫內容，然後刪除所有日誌紀錄。如果夠幸運，你可能查覺此次的資料外洩行為。圖 1-8 是整個入侵過程的示意圖。

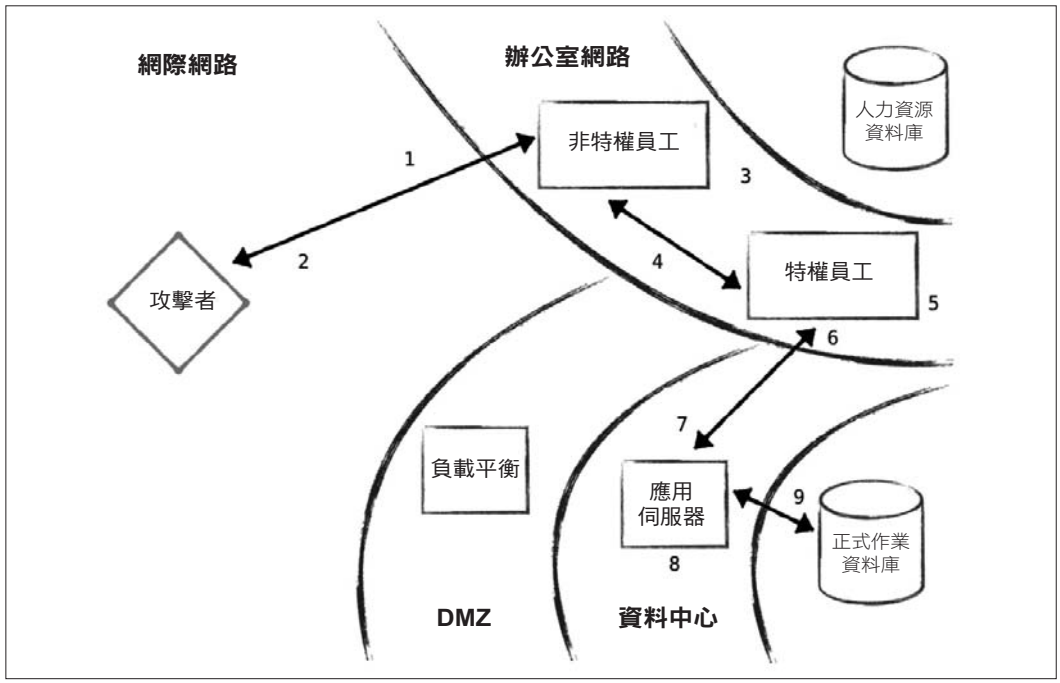

圖 1-8　攻擊者進入公司網路，然後入侵營運系統的網路環境。

同時存在固定和多變的性質

考慮如何以及採何種方式使用代理資料，必須了解其格式及找出特定資料的位置，某些資料的「坐標」必須是固定且公開，以確保整體控制平面系統的一致性，這種概念與關聯式資料庫類似，應用程式必須要知道資料庫的設計綱要，才能取得正確資料。

在實作和維護零信任控制平面系統時，資料相容性非常重要，零信任網路很可能利用許多系統和資料來源建構一個代理員，若沒有某種設計標準，不僅難以一致方式顯示資料，還會對引入新控制平面系統或代理資料的工作產生負面影響，這對成長中的零信任網路有絕對性影響。

有件事應謹記在心，由於資料來源端（如設備資產管理）無不可避免的純度問題，代理員所擁有的資料可能相當稀疏，基於某種因素，裡頭部分欄位可能未填入資料或填寫不完全，因此，只能力求代理員完美，卻難盡善盡美，與其追求資料純度（問題將會隨著規模的擴大而變得困難），不如認清資料不可能完整存在的事實，依照現實來制定政策才是最好的抉擇，儘管可能要求代理員要有某個特定資料，當缺少這項資料，應該具備以其他資料替代的思維。

尋求標準化

有人可能覺得資料格式似乎和使用它的機構密不可分，代理員的內容可能與商業邏輯的特徵或本機資訊相關，在這種情況下，代理資料的標準化可行嗎？

幸好這方面已經有一些標準出現了，最好的例子就是**簡單網路管理協定**（SNMP）及其使用的**管理資訊庫**（MIB）。

SNMP 是一種常用於網路設備管理的協定，允許設備以標準又靈活的方式向網管人員和管理系統公開資料，MIB 組件提供資料格式的定義及說明，這是一個稱為**物件識別碼**（OID）的資料集合，每個 OID 描述一個已在**國際標準化機構**（ISO）註冊的特定資料，為資料的特定部分提供可被廣泛接受的「坐標」。

圖 3-1 就是一組簡化節點後的 OID 樹結構：

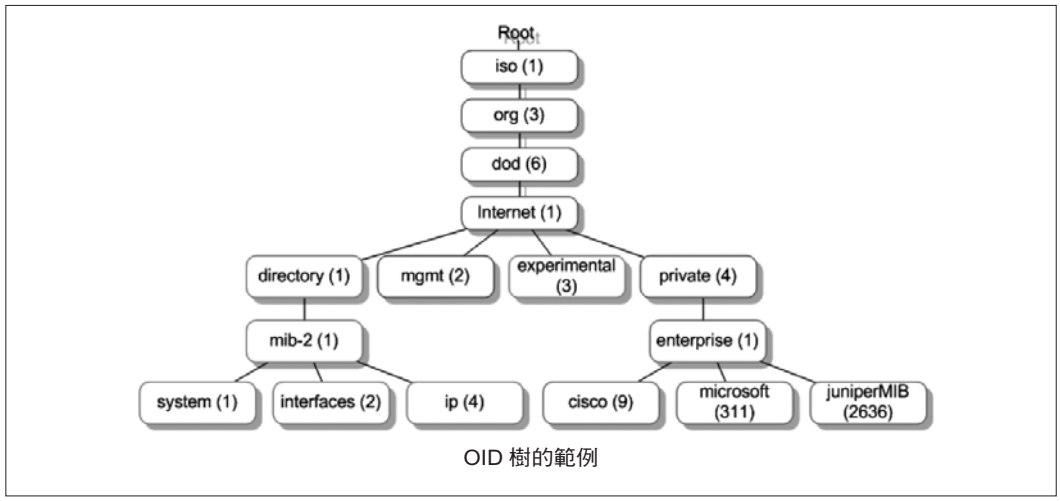

圖 3-1　簡化後的 OID 樹之節點組織圖

此範例中，「ip」節點和其相關資料可藉由 *1.3.6.1.1.1.4* 來定位，而 MIB 則為 OID 提供相關資訊。例如 Cisco 的 MIB 可能為 OID 樹在 1.3.6.1.4.1.9 底下部分提供所有 OID 的定義，包括適合人類閱讀的描述。

一般會為機構或製造商保留足夠的 OID 空間，因此，可以擴充此份註冊表。藉由 MIB 就可以將 OID 與 IP 位址進行比對，其中 IP 位址可代表全球唯一的某部特定主機，而 OID 則代表全球唯一的識別資料。

私有 IP 位址空間很適合應用在專屬或特定站台，可惜 OID 沒有適用於私有 IP 位址的機制，折衷作法是到**網際網路編號分配機構**（IANA；*http://pen.iana.org/pen/PenApplication.page*）註冊一組**企業專屬編號**（Private Enterprise Number），可以為你的機構提供專用的 OID 前綴代號，註冊企業專屬編號是免費的，只需回答一些問題而已。業界曾經打算仿照 IP 位址那般，規劃出私有的 OID 範圍，但並未成功。

雖然缺少供實驗或機構內部使用、真正免費／私有的 OID 空間，但在考慮代理的標準化時，SNMP 仍然提供實用的類比用法，它能提供資料格式描述和封包，以及如何將資料傳輸到另一個系統並被理解，而這些資料可使用其唯一的 OID 輕鬆找到和識別。

信任引擎

信任引擎是零信任網路中針對特定請求或活動執行風險分析的系統，它會將風險分析結果化作評估數值，政策引擎會利用此數值做出最終授權決策。

為了計算某個單元體（entity）的分數，信任引擎會從權威資產清單系統裡讀取資料，以便檢查該單元體的屬性。例如信任引擎可以從資產清單取得某設備最近一次稽核時點、或者它是否具備特殊硬體安全防護機制等訊息。

要建立風險評估的數值是一項艱鉅任務，一種可行的方法是定義一組評估單元體風險值的特定規則，例如，設備未安裝最新版的軟體修補程式就被扣分；類似方式，如果使用者不斷發生身分驗證失敗，就降低其信任分數。

特定的信任評分手法或許容易上手，但靜態定義的規則可能無法抵擋預期之外的攻擊向量，除了使用靜態規則外，成熟的信任引擎還會藉由機器學習技術來推導信任分數。

機器學習利用既有活動資料子集（稱為**訓練資料**）推導出評分函數，這些訓練資料是與受信任或不受信任單元體有關的原始觀測值，從資料中提取特徵用於推導出由電腦產生的評分函數。評分函數是機器學習所建立的模型，它會針對和訓練資料相同格式的資料進行處理，將所得到的分數與人為定義的風險評分進行比較，依照模型的能力，可不斷提升對所分析的資料之風險預測精準度，當模型有足夠準確度，就能預測未曾見過的網路請求之風險。

雖然愈來愈多以往不易處理的電腦問題都交由機器學習計算，但因推導評分模型的限制，或者機構需要自行定義評分規則，目前仍無法從信任引擎中排除定義明確的靜態規則，信任引擎通常會混合使用靜態評分規則和機器學習評分方式。

哪些單元體需要評分？

零信任網路的哪些組件需要評分？是每個單獨的單元體（使用者、設備和應用程式）、整個網路代理，還是兩者都需要評分？這是個有趣的問題。且來看一些情境。

想像一下，使用者的身分憑據遭受惡意第三方的暴力攻擊，某些系統是利用鎖定帳號的方式來減輕這種威脅的影響，但它有可能演變成對特定使用者的**阻斷服務**（DoS）攻擊。若只是依照使用者的負面行為進行評分，在零信任網路照樣會遇到相同問題，較好的作法是針對網路代理員進行身分驗證，因此便能對抗攻擊者的網路代理員，而合法使用者的網路代理員則不會受到傷害。在此情境下，就應該以網路代理員做為評分對象。

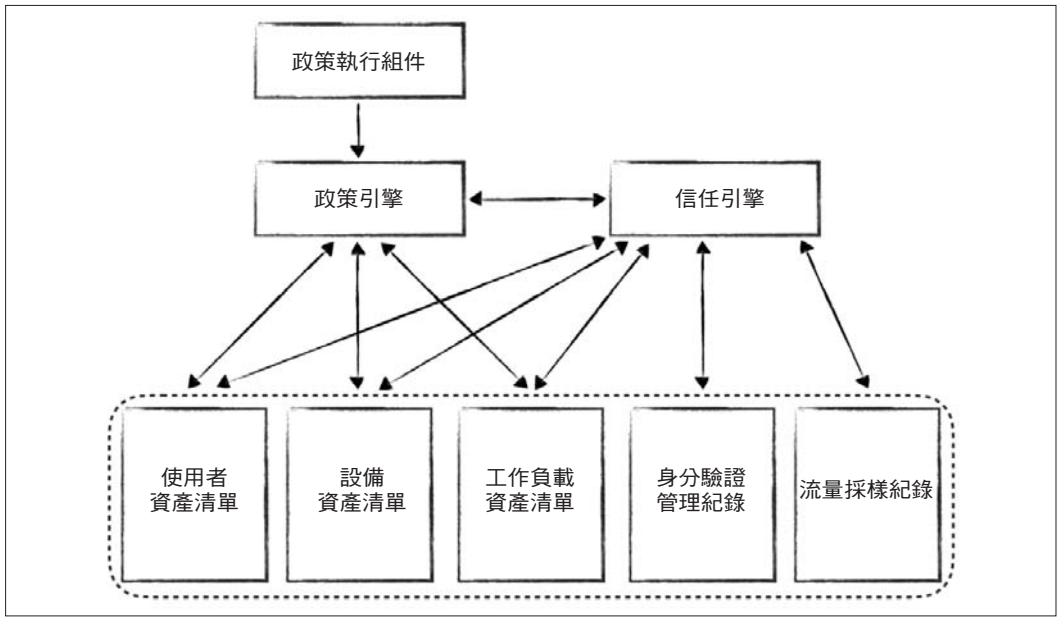

圖 4-4　政策引擎直接或間接經由信任引擎使用權威資料儲存系統

先前談到過信任引擎利用資料儲存系統來產生信任分數，作為政策引擎的決策依據，也就是說，資料儲存系統的資訊流向授權系統，最終提供給政策引擎進行決策。資料儲存系統除直接和間接供政策引擎使用，同時，對需要網路狀態權威資訊的其他系統也很有用處。

零信任網路傾向於按功能劃分成多個資料儲存系統，主要有兩種類型：**資產清單**（inventory）和**歷程紀錄**（historical）。資產清單是全體一致的真實來源，記錄資源（資產）的當前狀態，例如儲存所有使用者資訊的人員資產清單，或者記錄已知設備的資訊之設備資產清單。

資產清單裡有一個主鍵代表欲追蹤的單元體，以使用者而言，可選擇帳號作為主鍵；設備則可使用序號當主鍵。對零信任代理員進行身分驗證時，也是使用資產清單中的主鍵來代表它的身分，當要驗證使用者的身分時，政策引擎取得使用者的帳號，並完成身分驗證程序，然後，利用此帳號作為查詢資產清單的主鍵。牢記此流程及其目的，在實作系統及選用身分驗證方式時，將有助於選擇正確的主鍵。

這項特質正是 TPM 在零信任網路的設備身分驗證上受到重視的原因，一些出色的軟體式身分管理及驗證框架（如 X.509）為設備身分驗證付出相當心力，但無法將軟體金鑰與它要證明的設備硬體綁定，難以**真正**確立設備的身分，而 TPM 藉由硬體綁定，解決了這個問題。

使用 TPM 加密資料

TPM 會產生並儲存一對所謂的**儲存根金鑰**（SRK），該金鑰對代表 TPM 設備的信任根，凡使用此 TPM 公鑰所加密的資料，只有此 TPM 本身能夠解密。

聰明如你，一定會質疑這種功能是否勝任大量資料加密需求。執行非對稱加解密非常耗用電腦資源，故不適合批量資料（bulk data）加密場合，要利用 TPM 處理批量資料加密，就必須想辦法減少由 SRK 負責保護的資料量。

一種簡單的作法是：隨機產生加密金鑰（簡稱密鑰）供高性能的對稱加密演算法（即 AES）加密批量資料，然後再將此把 AES 金鑰交由 SRK 加密。如圖 5-2 所示，此策略可確保除了原本的 TPM 外，其他人無法取得密鑰。

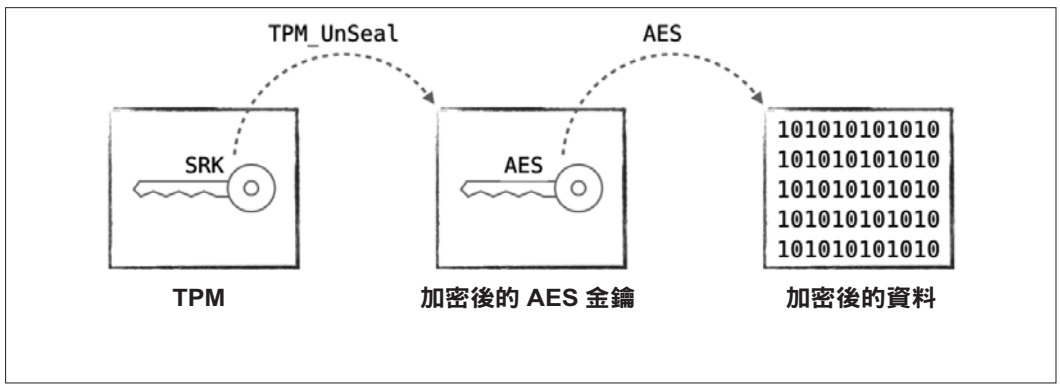

圖 5-2　使用 AES 金鑰進行資料加密，而 AES 金鑰又由 TPM 加密。

許多公開的 TPM 函式庫都提供執行上述作業所需的 API，在使用這些函式庫之前，筆者建議應先仔細驗核，確保其功能符合需求。

中介金鑰和密碼

使用 TPM 加密資料時，多數 TPM 函式庫（如 TrouSerS）會建立中介金鑰，也就是說，它們會要求 TPM 另外建立一對新的非對稱金鑰，並以公鑰加密 AES 金鑰，最後再使用 SRK 加密私鑰。在為資料解密時，必須先將中介私鑰解密，再用它為 AES 金鑰解密，之後，才能解出原始資料。

這種實作方式似乎有些奇怪，卻有存在的理由，額外增加的中介層，可讓發送加密資料的作業更具彈性。SRK 和中介金鑰都能夠為密碼加密，透過中介金鑰便能引入其他密碼，或許更多人知道這種作法。

不管這種方式對於你的部署有無意義，若基於「加密金鑰只能在此設備上解密」的目標，就算不使用中介金鑰也可以，甚至處理效率可能更高。

支援 TPM 安全儲存的應用重點是保護設備的 X.509 憑證之私鑰，此私鑰具有證明設備身分的權威性，如果私鑰被偷，就等同身分被偷。使用 TPM 加密 X.509 的私鑰，雖然駭客仍能從磁碟取得加密後的金鑰，但沒有原來的硬體，還是無法還原使用。

金鑰仍可能被偷

將設備的私鑰加密，再利用 SRK 封裝加密用的密鑰，這樣做並不能解決偷竊攻擊，它可以防止金鑰直接從磁碟讀取出，若攻擊者能夠提權，在高特權之下仍可能從記憶體讀取密鑰，或者要求 TPM 執行解密。

接下來的兩小節將提供驗證硬體身分的其他方案。

平台組態暫存器

平台組態暫存器（PCR）是 TPM 的重要功能之一，它提供多個儲存槽，用來儲存執行中軟體的雜湊值。首先從 BIOS 的雜湊值開始存放、接著是啟動紀錄、組態資訊等等的雜湊值，依序儲存的雜湊值，之後可用來證明系統的組態或狀態未被竄改。底下是儲存在 TPM 的前幾個暫存器的示例：

```
PCR-00: A8 5A 84 B7 38 FC ...        # BIOS
PCR-01: 11 40 C1 7D 0D 25 ...        # BIOS 的組態
PCR-02: A3 82 9A 64 61 85 ...        # 選用唯讀記憶體（ROM）
PCR-03: B2 A8 3B 0E BF 2F ...        # 選用唯讀記憶體的組態
PCR-04: 78 93 CF 58 0E E1 ...        # 主要開機磁區（MBR）
PCR-05: 72 A7 A9 6C 96 39 ...        # 主要開機磁區的組態
```

資產清單管理

驗證設備身分的有效性和完整性，已向強健零信任安全邁出一大步，但能夠驗證設備身分只是所面臨的挑戰之一部分，我們還需要許多資訊才能計算政策，以及做成決策。

資產清單管理涉及設備和其屬性的分類登錄及管理，維護伺服器和終端設備的資產清單紀錄是同樣重要，有時將它們看作網路單元體（entity）而非實體設備，或許更有幫助，雖然它們真的是實體設備，但也可以是網路上的邏輯單元體。

例如，依照需要，可將虛擬機或容器想像成「設備」，畢竟，它們和真實伺服器相同，具有許多類似表達的屬性，若將所有虛擬機的流量與承載虛擬機的主機之流量混為一談，而將它們歸結到主機的安全政策上，會將我們推回邊界安全模型；反之，零信任模型提倡追蹤工作負載，以推動制定所需的網路政策，在這種情況下，應該針對資產（或工作負載）資料庫進行處理，以適應虛擬化／容器化環境所面臨的高速變動，傳統的資產清單管理系統和工作負載調度程式或許是不同系統，但還是可以協同作業。如圖 5-3 所示，就本書的目標，工作負載調度服務也可以當作某種類型的資產清單管理系統。

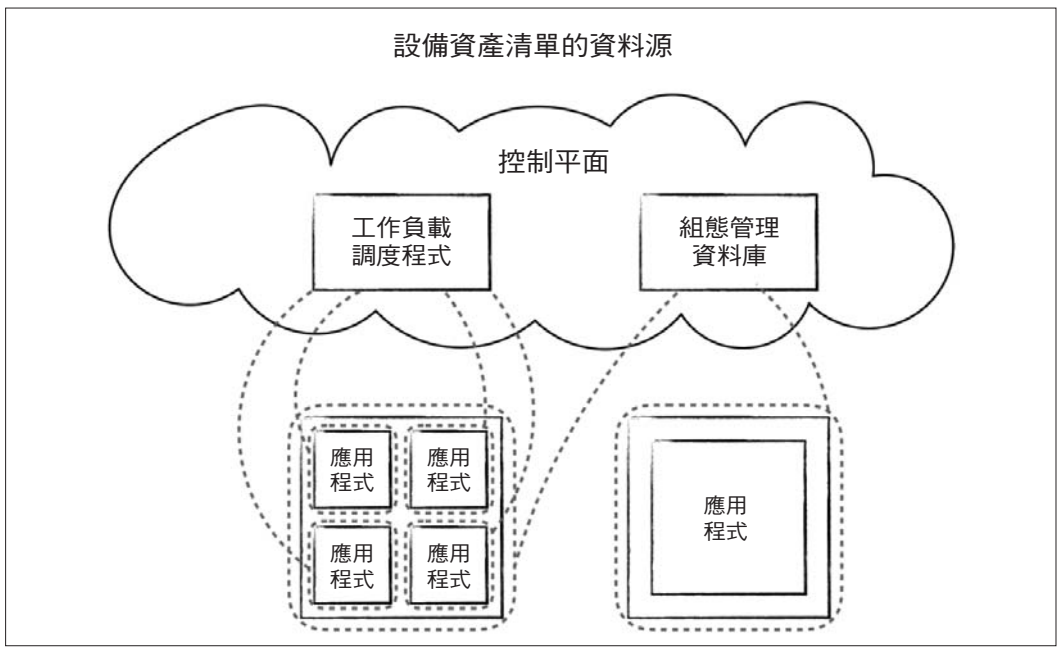

圖 5-3　以工作負載調度程式和組態管理資料庫當作控制平面的資產清單資料庫

基於這個原因，輪換作業就很重要，前面花了一些篇幅談論輪換原則的重要性，就算設備也適用。當然，對「設備」有不同界定時，「輪換」的方式顯然也會不一樣。如果是在雲端架構上運行的設施，則「設備」可能是一個主機實例（軟體機器），輪換作業就簡單多了，只需要卸載舊實例（instance），再建一個新實例即可（透過組態管理就做到了，不是嗎？）如果是實體設備，作業起來就有些麻煩。

重建映像檔是邏輯上輪換設備的好方法，這是相當低階的作業，可以有效消除當今已知的大量持久性威脅，我們傾向於信任一部新建映像的設備，而不是一部已運行一年的老設備。雖然重建映像檔不能解決圖 5-4 所示的硬體攻擊或其他低階系統攻擊，但對於難執行實體設備輪換的地方，這是可以接受的折衷方案。資料中心和供應鏈安全可以稍稍減低這種憂慮。

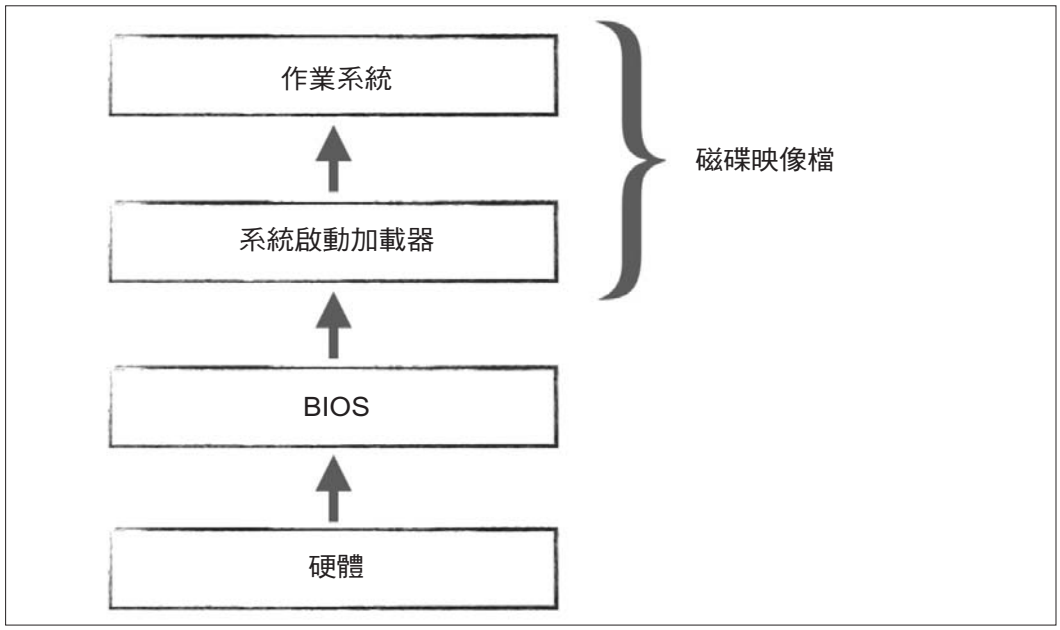

圖 5-4　磁碟映像檔可解決大多數已進駐系統的惡意軟體，但無法全數清除。

在管理用戶端設備時，情勢就大不相同，重新建立用戶端設備的映像檔，對使用者是非常不便的，隨著時間演進，使用者可能自行調整機器組態或其保有的資訊，又無法有效或安全留存這些客制變更，通常，配送新設備給他時，他會想要將舊的映像檔倒回來！對於試圖保護用戶端設備的人來說，這絕對不是一個好消息。

掃描結果可以和已知的不良特徵值（如惡意軟體或有漏洞的合法軟體之版本）進行交互參照，進而產生類似圖 5-5 所示的報告，從檢測到的已知不良特徵值，會適時地影響設備的信任度。

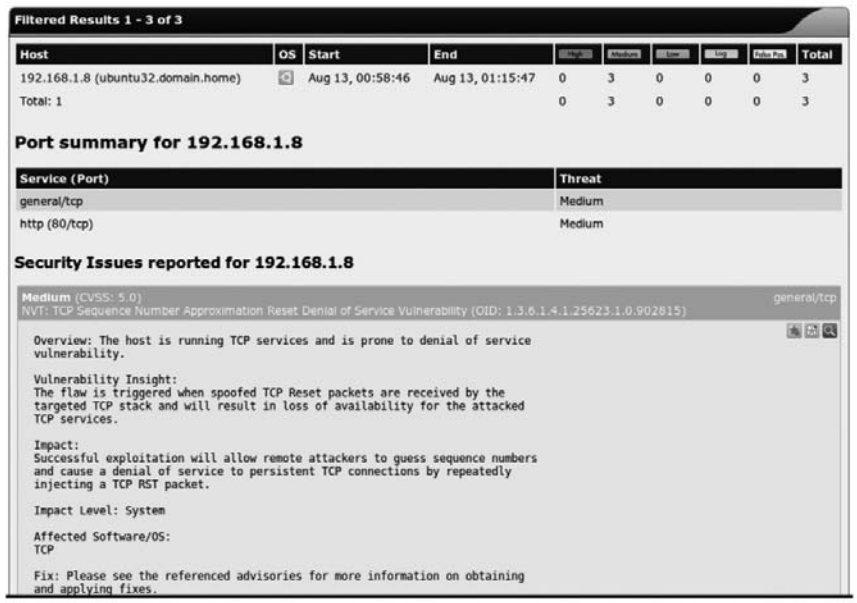

圖 5-5　OpenVAS 的 Greenbone Web 使用者界面顯示從掃描目標找到三個「中風險」漏洞（插圖取自：*https://www.flickr.com/photos/xmodulo/9499759166*）

有許多開源和商業類型的漏洞掃描工具可供選用，包括 OpenVAS、Nessus 和 Metasploit，這些工具都已相當成熟，並受到許多機構使用。

不幸的，漏洞掃描與本機測定會遭遇相同的根本問題：依賴對端點質詢或探測。不同的是：本機測定「詢問」某人是否搶銀行；漏洞掃描則「觀察」某人是否搶銀行。當然，有時被捉的搶匪會承認犯行，但高竿的匪徒絕不會因此露出馬腳。捕捉現行犯會比事後追查來得有效率得多。解決此難題的更多訊息，可參考本章後面的「網路通訊態樣」小節。