瞭解並選擇 Secrets 資訊管理平台

1.0 版
發佈日期：2018 年 1 月

Securosis 有限責任公司 515 E. Carefree Highway Suite #766 Phoenix，AZ 85085 T 602-412-3051
info@securosis.com www.securosis.com

作者附註

本報告的內容以獨立於任何贊助商的方式撰寫。具體內容基於 Securosis 部落客中最初發佈的資料，但經過增強、審核與專業編輯。

特別感謝 Chris Pepper 全力提供編輯與內容支援的協助。

CyberArk 提供授權

CyberArk（NASDAQ：CYBR）是特權帳號安全領域的全球領導者——特權帳號安全是在整個企業、雲端與 DevOps 流程內保護資料、基礎架構及資產的 IT 安全性的關鍵領域。CyberArk 提供業界最全方位解決方案，可降低特權憑證或帳密與秘密資訊帶來的資安風險。本公司深得全世界領先企業的信賴，包括 50% 以上的財富前 100 大公司。這些公司利用 CyberArk 產品來防止外部攻擊者與心懷惡意的內部使用者所發起攻擊。CyberArk 是一家全球性公司，總部設在以色列 Petach Tikva，美國總部位於麻塞諸塞州 Newton。公司還在美洲、EMEA、亞太地區及日本設有辦事處。有關 CyberArk 的更多資訊請瀏覽 www.cyberark.com，閱讀 CyberArk 部落客或關注 Twitter（@CyberArk）、LinkedIn 或 Facebook。

版權

瞭解並選擇 Secrets 資訊管理平台

目錄

瞭解並選擇 Secrets 資訊管理平台 1

作者附註 2

CyberArk 提供授權 2

版權 2

引言 4

機器身份的挑戰 5

應用場景 7

主要客戶應用場景 7

特性與功能 9

核心功能與特性 9

進階功能特性 12

部署考慮事項 14

產品類別 14

部署模式 15

總結 17

關於分析師 18

關於 Securosis 19

引言

顧名思義，Secrets 資訊管理平台負責保存、管理與提供 Secrets 資訊。此技術可解決大多數資安管理人員還不知道的幾大問題。在開發團隊越來越多地利用自動化與協作流程技術的過程中，也必須解決新的資安問題。我們對雲端服務自動化的依賴，使應用程式與 IT 服務變得越來越快，越來越有彈性；但伴隨此項優勢而來的還有為機器與軟體（而不僅是人）提供存取權的需求。從某種觀點，這是自本世紀初以來身份管理問題的新的重複，只是在個人設備與雲端運算基礎架構中被再次放大。

此問題的起源是：開發人員實現軟體建置與測試的自動化，而 IT 部門正實現佈建工作的自動化；這兩個團隊都仍然認為，安全性減緩他們前進的步伐。連續整合、連續部署與 DevOps 實踐都可以協助提高靈敏性，但同時也造成資安風險——包括將 Secrets 資訊保存在原始程式碼庫中以及隨意將憑證或帳密保存在各處。這種不好的習慣使生產環境中的每種軟體皆可能引發風險！

所有軟體都需要憑證或帳密才能存取其它資源、與資料庫通信、獲取加密金鑰或存取其它服務。但是，這些存取特權必須得到嚴密保護，防止被攻擊者濫用！此問題需要您全面瞭解應該分配什麼權限以及軟體可以接受什麼格式，然後在某使用者不直接參與或不能直接參與的情況下安全分配存取權。開發人員負責與身份來源（如目錄服務）的整合，但開發人員通常不瞭解目前可協助他們分發憑證或帳密給計畫中*目標人員*的技術。

為了應對這些變化，業界開發強大的工具與平台來保護敏感性資料並安全地分派權限。業界將這種全新類別的產品稱為「Secrets 資訊管理」產品；此產品正改變著我們提供身份、Secrets 資訊及權杖以與驗證系統自動建立信任關係的方式。實際上，Secrets 資訊管理是有助於從 DevOps 轉型到 SecDevOps 階段的核心組成部分。本次研究將深入討論為什麼說這對很多公司來說皆是重大問題、這些新平台可以解決哪些類型的問題以及這些新平台如何在更新環境中運作。

機器身份的挑戰

要使自動化腳本正常執行，取得 Secrets 資訊是關鍵；但很多公司堅持採用傳統 (簡單的) 操作模式：將 Secrets 資訊保存在檔案中或嵌入到腳本中，以確保在不需要人為干預的情況下完成任務。開發人員非常清楚這是暫時被隱藏的問題，而且是會快速蔓延的技術與資安債務，但是只有在稽核過程中或發生資安事件後才會顯露出來。開發人員當然*不會*刻意主動地告訴資安部門關於他們處理 Secrets 資訊的方式，因此大多數首席資訊資安長 (CISO) 與資安架構師不瞭解這個新問題，直到發生資安事件。

這不是新出現的問題。幾十年來，管理者們一直將 Secrets 資訊保存在不安全的檔案中。沒有哪位管理者願意被半夜叫起來工作，輸入密碼以便啟動某應用程式。IT 管理者經常將加密金鑰保存在檔案中，以便作業系統或應用程式可以在需要時存取。資料庫管理者將加密金鑰與密碼保存在檔案中以方便完成自動重啟。或者，在企業網路遭受到更嚴重的攻擊，導致這種業界普遍採用的做法被禁止之前，管理者會一直這麼做。自那以後，我們一直依賴從手動干預到金鑰管理服務甚至硬體鎖的一切來作為信任之根源 (root of trust)，進而建立身份並佈建系統。但是，這種老方法不能在新的運算與開發環境中提供所需的安全性；由於我們提供軟體與服務的方式的動態性，導致最終結果要嚴重得多。

聽起來確實是陳腔濫調，但 IT 與應用程式環境正真正經歷重大變化。作為微服務部署應用程式以及部署到容器中的新方法正協助我們更具成本效益地擴展服務與大型系統。軟體定義的 IT 堆疊與透過 API 實現的精細服務控制可帶來明顯的敏捷性優勢。連續整合 (Continuous Integration) 與 DevOps 等現代維運模式使這些優勢更加明顯，可以更快速更可靠地將應用程式與基礎架構推向市場。

或許，目前影響軟體發展與 IT 的最重大變革是雲端運算。雲端服務具有隨需與彈性的特點，將帶來明顯的優勢 (這是根據定義為軟體的自動化基礎架構所預測的)。雲端不是其它技術進步的必要條件，但雲端使技術更加強大。同時利用所有這些技術進步成果，只需幾行程式碼就可以在幾分鐘內啟動或關閉整個 (虛擬) 資料中心，同時最大幅度減少人為干預或工作。

在帶來種種優勢的同時，自動化與協作流程也帶來新的資安問題。目前的主要問題是如何安全的共用 Secrets 資訊。在「機器」不再是託管機櫃內安裝的硬體設備，而是一個短暫實例，或者是同時執行的*數千*個實例（或許根據需求的瞬間變化而大量建立與部署）時，此項問題顯得尤其嚴重。我們如何追蹤具體伺服器、虛擬機器或容器，或者我們應為伺服器、虛擬機器或容器分別賦予哪些權限？隨著我們在不同團隊間擴大自動化與協調作程範圍，此問題的影響範圍也越來越大。開發團隊需要在不同團隊間共用資料、組態設定與存取金鑰，以合作完成應用程式開發與測試。自動組建伺服器（Build Server）需要存取原始碼版控系統、API 閘道與使用者角色以完成任務。伺服器需要存取加密磁碟，應用程式需要存取資料庫，而容器從一開始啟動就需要得到適當的特權。自動化服務不可能等待使用者鍵入密碼或提供憑證或帳密！因此，我們需要新的、敏捷的自動化方法來提供資料、身份與存取權。

應用場景

需要 Secrets 資訊管理的原因有很多，應用場景也是五花八門。即便如此，要解決的主要問題是如何安全地為服務設定存取權，這是目前非常缺乏的。相反，Secrets 資訊一般以純文字方式保存在文件與應用程式中。隨著大企業逐步大規模採用這些更敏捷的部署方法，這些問題變得更加迫切。大多數公司已經開始依賴身份儲存系統 (Identity Stores Systems) 來集中控制身份與存取權。然而缺乏一種合理的分配機制，以同等地支援大企業內混合 / 複雜環境中的安全性原則，包括公司內與公有雲端環境。

主要客戶應用場景

1. API **閘道與存取金鑰：**應用程式設計介面 (Application Programming Interfaces) 是軟體程式與其它軟體及服務進行互動的途徑。這些 API 是實現協調操作的基本介面。要使用 API，您必須首先向 API 閘道完成您自己或您的程式的身份驗證。這一般透過提供存取金鑰、權杖或回答密碼詢問的方式來完成。為簡化自動化，許多開發人員以寫死的方式產生存取金鑰，因而在面對簡單的檔案或程式碼檢查時不堪一擊。而且常見的情況是，即使保存在開發人員桌面上的專用檔案中，金鑰也會偶爾被共用或公佈——有時是被新增到共用程式碼儲存庫中，這樣做的目的是為了對存取金鑰進行保密，同時在需要時提供給執行中的應用程式。

2. **服務：**應用程式作為獨立實體的情況非常少。應用程式一般包括多種不同的元件、資料庫與支援設備。在現今的應用程式架構中，我們會啟動某應用程式的多個執行個體來確保延展性與彈性。在我們啟動應用程式時，不管是在容器中還是在虛擬機器或伺服器上，我們都必須佈建應用程式的組態設定資料、身份憑證與權杖。新建立的虛擬機器、容器或應用程式如何知道自己的身份並存取所需的資源？我們如何從大量相同的容器中正確地找出特定容器？在實現全面環境自動化的競賽中，各企業的自動化步伐太快，反而欲速則不達且承受更多風險，無法在安全性與開發速度之間實現正確的平衡。開發人員通常將憑證或帳密保存在設定檔中；而應用程式與伺服器啟動後就可以方便地獲得這些設定檔。我們發現，生產環境中的憑證或帳密經常與品保及開發系統共用，這通常很不安全而且不受任何監控。本不應該擁有存取權的其它應用程式與服務也通常共用這些憑證或帳密。我們的目標是隔離憑證或帳密而不導致破壞或不可接受的障礙。

3. **開發自動化：**大多數軟體發展環境都不安全。開發人員認為開發環境中的資安措施會降低他們的工作速度，因此開發人員在開發過程中經常繞過資安控制措施。開發環境通常在開發人員的控制下，並在開發部門的伺服器上執行，因此外部人員很少知道開發環境在哪裡以及如何執行。10 多年來，一夜之間製造出來的伺服器比比皆是，伺服器的自動化程度不斷提高以改進敏捷性。隨著速度的提高，一切不再受到人們的監控。隨著新程式碼、開發模版與腳本經過檢查並新增到儲存庫中，Jenkins 與 Bamboo 等組建伺服器會自動重新產生應用程式。但是，組建伺服器也會驗證新增的內容，進行品質保證與資安測試，而且在測試失敗的情況下會終止此項工作。這意味著品質測試與資安測試成為整個開發流程的一部分，而且不會再減緩開發流程，反而可以作為一道防護措施，阻止不良程式碼進入到生產環境中。將 Secrets 資訊新增到開發流程中，也是自動化流程的一部分，而且與開發及發佈應用程式需要同樣的敏捷性與自動化。我們可透過規則來定義應賦予的權限，確保可以安全地交付程式碼與服務而不需要人為干預。

4. **佈建機器身份：**在利用雲端服務的過程中，我們對「伺服器」的定義也在發生變化。雲端的使用概念是將運算、網路與儲存作為服務來隨需分配，然後在不需要時撤銷。因此，我們過去在公司內視為「機器」的項目現在變成短暫的執行個體（通常是虛擬的而且數量很多），作為伺服器鏡像、容器或類似概念。但是，由於我們連續不斷地增加與減少這些「機器」的數量，因此很難追蹤哪台伺服器回應哪項請求，因此我們需要有效的方法來為「機器」分派唯一的身份。我們可能需要找到可能未正常運作或受到攻擊的機器執行個體並採取事件回應措施，但我們不能針對所有執行個體採取千篇一律的補救措施，因此以唯一的身份為目標至關重要。

5. **加密資料：**提供加密金鑰來解鎖加密的磁碟區與檔案儲存是一項常見任務，不管是在公司內還是對於雲端服務皆如此。過去，我們使用金鑰管理伺服器來安全分派與管理金鑰，但很多商用金鑰管理工具（包括硬體與軟體）尚未改進，不適合 IaaS 或 PaaS。此外，開發人員現在需要更緊密的 API 整合，以確保無縫地搭配應用程式使用。企業內經常缺乏此項功能，因此某些團隊使用雲端原生的金鑰管理，而另外一些則選擇用 Secrets 資訊管理來代替。

6. **共用：**協作軟體有助於開發、品質保證與產品管理團隊展開專案合作——即使人們遠距工作或企業團隊遍佈不同的地方。在某些情況下，問題是如何在由遠端開發人員組成的團隊中安全地共用資訊，或在多個資料中心間共用 Secrets 資料而不以純文字形式洩露出去。用於保存聊天與協作服務資料的資料庫一般不是非常安全，向協作工作者發放文字憑證是一種絕望的作法。該解決方案需要一種強大的集中儲存庫，可供選定的使用者群組儲存與搜索 Secrets 資訊。

當然還有大量其它應用場景。在採訪中，我們討論從簡單密碼到比特幣錢包的一切場景。但在本調查中，我們需要將精力集中在開發人員與 IT 資安人員提出的問題上。

特性與功能

現在我們來討論 Secrets 資訊管理平台的核心功能與特性。Secrets 資訊管理平台是一種全新的產品，因此市場上存在很少功能特性一致的產品。其中很多被優雅地稱作「產品」，實際上與個人辦公工具沒有多大區別。功能全面的企業級平台是個例外。然而，所有 Secrets 資訊管理平台都需要提供一些基本功能，包括 Secrets 資訊的安全儲存、佈建 Secrets 資訊、身份管理及可整合的 API。在考慮您需要平台提供哪些功能時，須要記住的重要一點是，平台最初是為了執行某一種具體任務而開發的——如在執行時將 Secrets 資訊插入到容器中、實現與 Jenkins 組建伺服器的緊密整合或輔助雲端身份服務。這些解決方案可以很好地完成某一種任務但很少適合多種應用場景，而且會導致多個不受管理的資安孤島。

現在讓我們更深入地看看主要功能與特性。

核心功能與特性

Secrets 資訊管理平台的設計旨在支援其它應用程式的軟體應用程式，負責一項非常重要的任務：安全保存 Secrets 資訊並且發送給正確的人或應用程式。Secrets 資訊管理平台的最重要特徵是，*在任何情況下都不應將 Secrets 資訊以純文字形式保存在各處！*安全儲存是首要重點。

保存 Secrets 資訊

我們評估的每一種工具幾乎都可以提供一個或多個加密儲存庫（很多產品稱作「金庫」(Vault)）來保存 Secrets 資訊。在您將 Secrets 資訊新增到儲存庫中或更新 Secrets 資訊時，Secrets 資訊會在被寫入到存放裝置中之前自動完成加密。雖然聽起來有些令人震驚，但我們評估的產品中確實至少有一種不對 Secrets 資訊進行*加密*——相反，Secrets 資訊以純文字形式保存。

這當然不能把不提供加密的產品列入考慮範圍之內。幸運的是，大多數金庫都以經過驗證方式來實作眾所熟知的演算法，對 Secrets 資訊進行加密。但是，您應根據自己的具體監管與合約要求來對任何實作方法進行驗證，確保金庫滿足您的資安要求。

除了提供「短暫 Secrets 資訊」（將於下文更詳細介紹）的一些平台外，所有 Secrets 資料都保存在這些儲存庫中供將來使用。任何資料都不會以純文字形式保存。每種平台將 Secrets 資訊與特定使用者身份識別符號、憑證、帳密及角色相關聯的方式天差地別。每種平台都有自己的方法來管理 Secrets 資訊，但平台一般都使用獨特的識別符號或鍵值對 (key-value pair) 來識別每項 Secrets 資訊。

某些產品會保存每項 Secrets 資訊的多個版本，以確保必要時可以按時間追溯過往修改及操作，但詳細資訊屬於各產品獨家秘方的一部分。

不同產品採用的儲存庫結構也有天壤之別。某些產品將資料保存在簡單文字或 JSON 檔案中。某些產品在 NoSQL 型資料庫中使用鍵值對。一些產品使用關聯式資料庫或 NoSQL 資料庫。還有一些產品同時利用多種儲存庫類型來更清楚地隔離 Secrets 資訊與 / 或應用場景。儲存庫結構主要根據安全性來決定的情況非常少——其它影響因素包括降低成本與對產品開發人員而言較為簡便易用。儘管任何類型的儲存庫都可能很安全，但儲存庫的選擇會影響延展性、可用性以及搜尋與設定 Secrets 資訊的速度。

另一應考慮事項是儲存庫可以處理哪些資料類型。我們評估的大多數平台可以處理您希望保存的任何類型的資料：字串值、文字欄位、N-tuple 對與二進位資料。索引編製工作通常在您插入項目的同時自動完成，以提高後期的查找與檢索速度。某些平台確實只能處理字串，這樣可以簡化 API 但同時也限制使用範圍。此外，針對特定應用場景定制的產品可能不適合其它應用場景或不同團隊。

身份與存取管理

與外部 Active Directory 或 LDAP 服務相比，大多數 Secrets 資訊管理平台優先使用 IAM；其來有自，因為大多數公司目前已經部署 IAM 基礎架構。使用者向目錄存放庫 (directory store) 進行身份驗證以獲取存取權，而伺服器利用現有的角色來決定授權該使用者存取哪些功能與 Secrets 資訊。大多數平台還能使用協力廠商雲端身份服務（Cloud Identity Service) 或特權存取管理服務，或直接整合雲端原生目錄服務。

介面與使用

大多數平台提供一個或多個程式設計介面。在自動化環境中處理 Secrets 資訊的最常用方法是存取 API。提供數量較小而且簡單的 API 呼叫 (API calls) 來進行連線身份驗證、插入記錄、查找 Secrets 資訊、或與特定使用者或服務共用的 Secrets 資訊。更先進的解決方案還允許透過 API 存取進階功能及管理功能。

命令列介面存取也很常見，在以命令驅動的 UNIX/Linux 環境中利用相同的基本功能。多種工具也直接或間接地提供圖形化使用者介面——有時透過另一開放原始碼專案。

共用 Secrets 資訊

Secrets 資訊管理系統的最有趣方面是如何與使用者、服務或應用程式共用 Secrets 資訊。您如何安全提供 Secrets 資訊給目標接收方？您如何在身份庫與需要 Secrets 資訊的服務之間建立信任關係？與前面討論的儲存庫一樣，Secrets 資訊在發送過程中必須受到保護——這通常意味著需要加密。您可以使用雙向身份驗證，透過多種不同的方式來安全發送 Secrets 資訊。讓我們看看這些常用方法。

- **加密的網路通訊：**經過身份驗證的服務或使用者通常透過加密連線以純文字形式取得 Secrets 資訊。某些使用 SSL，但這不是最理想的，我們建議盡可能避免使用這些平台。幸運的是，大多數平台使用最新版本的 TLS 傳輸層加密 (Transport Layer Encryption) ——這種方法在接收方與 Secrets 資訊管理伺服器間提供雙向身份驗證功能。利用 TLS 時，這些平台內建基本的憑證授權 (Certificate Authority) 功能。儘管不是成熟的 PKI 伺服器；但這些平台的內部 CA 意味著這些平台可以建立、發放與撤銷憑證；因此可在 Secrets 資訊共用系統 (如 Docker Swarm、Kubernetes Pods、Cloud Autoscale Group 等) 中用作密碼身份 (cryptographic identities) 的集中授權機構。

- **PKI：**幾種 Secrets 資訊管理平台組合外部身份管理與公用金鑰基礎架構 (Public Key Infrastructure)，驗證 Secrets 資訊接收方並發送 PKI 加密的資料封包。該平台可以決定誰將接收到 Secrets 資訊，然後利用接收方的公共金鑰對內容進行加密，以此確保唯有目標接收方可以利用自己的專用金鑰來解密 Secrets 資訊。還可以減輕甚至徹底消除加密所有網路通訊的需求。

- **暫存檔案：**使用容器時，很常見的一種情況是透過將 Secrets 資訊放在 UNIX 環境中的揮發性記憶體檔案系統 tmpfs 中來共用這些 Secrets 資訊，因此 Secrets 資訊管理伺服器可以將 Secrets 資訊保存到相同硬體上託管的容器中。對這些資料的存取權僅限於單一實體系統中的應用程式而言，由於資料只保存在記憶體中，所以存取速度非常快而且伺服器被移除時 Secrets 資訊會消失。不好的一面是這些 Secrets 資訊通常以純文字形式保存，因此操作時需要非常謹慎，須確保只有獲得授權的容器並設定命名空間來防止未經授權應用程式讀取 Secrets 資訊。如果有惡意程式碼進入實體主機上的任何容器中，這種模式就會失去作用。

- **封裝 (Wrapping)：**某些商用平台與雲端廠商在本地端利用對稱式金鑰密碼編碼，在每次啟動一種服務或代理程式 (agent) 時佈建一個新的唯一的金鑰。與 PKI 場景相似，這種情況下 Secrets 資訊也會根據需要使用的接收方的金鑰進行加密 (或封裝)，然後以加密型態發送。這種金鑰是短暫的，與雲端服務一樣會在代理程式或服務終止後丟棄。

- **插入：**在某些情況下，可以自動提供 Secrets 資訊。啟動虛擬伺服器時，Secrets 資訊可能是來自啟動時所提供的設定檔。Swarm 或 Pod 中的容器可能會被分配到一個身份憑證，藉此獲得存取權與特權。在這種模式下，Pod 或 Kubelet 中的每個容器會共用相同的 Secrets 資訊。此方式的目標是緩解進入環境的非法程式碼所帶來的風險，並自動獲得對 Secrets 資訊的存取權。

- **純文字：**是的，就是未加密的純文字。我們當然不能推薦不能保護 Secrets 資訊而是以純文字形式共用這些 Secrets 資訊的 Secrets 資訊管理系統。對大多數企業來說，這是無效的，因為這無法確保 Secrets 資訊的保密性。但是您需要瞭解的是，這種情況仍然存在——Secrets 資訊既透過網路傳輸，又在通用目錄或檔案中共用，雖然我們可以假設這些目錄或檔案本身是安全的。如果您認識到了這一點，就應尋找不同的產品。

進階功能特性

隨著 Secrets 資訊管理需求的變化，我們開始看到市場上出現商用 Secrets 資訊管理產品。這些平台的設計旨在支援我們前面討論過的多種主要應用場景，而且一般提供更先進的功能特性，如深度日誌建立與整合選項、與 IAM 服務的更緊密整合、Secrets 資訊產生以及 Secrets 資訊撤銷。隨著此項市場領域的不斷成熟，我們開始看到更先進的功能特性與更緊密的服務整合，進而減少您所需的黏合用程式碼 (glue code) 。下文列出一些我們曾遇到的進階功能特性 (未按特定順序) 。

- **管理 Secrets 資訊：**對任何 Secrets 資訊管理平台來說，授權模式概念都有效。身份管理與目錄服務一般作為「記錄系統」執行，其中規定哪些身份的使用者可以讀取或修改、更新或刪除身份。區分企業級工具的一個方面是分配管理 Secrets 資訊與管理 Secrets 資訊存取權限的能力。日誌、儲存、Secrets 資訊建立、恢復與容錯移轉設定正成為企業 Secrets 資訊管理平台的必要配備，且只應透過管理介面存取，而不是暴露給一般系統使用者。這是很重要的組成部分，因為 Secrets 資訊可根據使用情況設定安全性原則的能力——而這通常是監管或內部風險與合規性原則 (如定期更換與職責劃分等) 的強制性要求。允許機器或服務從存放裝置中讀取資訊但不能寫入或刪除資料的能力是另一個例子。隨著我們越來越依賴自動化服務，Secrets 資訊管理介於手動操作與自動化服務之間的可能性越來越大，包括建立、更新與強制實施規則。

- **佈建機器身份：**自動化是 Secrets 資訊管理目前成為開發與 IT 環境中的基本功能特性的根本原因。自動化必須能夠安全啟動服務、容器或伺服器，並且精確識別在相同群組中的服務、容器或伺服器。雖然聽起來有些令人震驚，但某些產品確實不能提供此項基本功能特性，或需要將 Secrets 資訊發送到共用檔案系統中，而不是直接發送唯一的 Secrets 資訊至每個機器或執行個體。您可能需要仔細瞭解如何建立與佈建機器身份。

- **Secrets 資訊建立：**Secrets 資訊管理平台現在能夠建立與發放 SSL 憑證、密碼、TLS 憑證、身份權杖、加密金鑰及其它有用的項目。在某些情況下，這些 Secrets 資訊可能有「有效期」以支援短期存取——逾期後 Secrets 資訊將不再有效。

- **撤銷：**這可以有助於 Secrets 資訊管理系統使憑證無效或撤銷。此功能特性一般在 Secrets 資訊管理器系統中提供，同時也可作為身份庫或憑證認證機構 (Certificate Authority) ——如容器協作流程環境，因此可以撤銷用戶端與其它使用者及服務進行通信的能力。

- **短暫的 Secrets 資訊：**容器、伺服器與 IaaS/PaaS 功能等都基本上是短時間內有效的。透過啟動多個應用程式執行個體並取代執行不正常的任何執行個體，就可以提供彈性。此概念同樣適用於安全性，基本理念是佈建的 Secrets 資訊像雲端伺服器一樣只在短時間內有效。我們可以根據需要來為伺服器執行個體或容器類別產生新的短暫 Secrets 資訊。如果 Secrets 資訊丟失或容器故障，我們可以隨需產生新的 Secrets 資訊。這對於身份憑證、加密金鑰及不同服務間共用的其它類型的 Secrets 資訊非常有用。這還有助於完成 Secrets 資訊與金鑰輪換，達到合規性要求。這些 Secrets 資訊不是長期保存的——而是，Secrets 資訊管理器可以動態地記錄為哪些服務分配短期有效的 Secrets 資訊。

- **加密即服務：**某些 Secrets 資訊管理平台根據請求來加密酬載 (payload)。簡單的 API 呼叫可以傳遞獨特的識別符號給酬載：將使用的加密金鑰或目標接收方，Secrets 資訊管理平台，作為加密引擎執行。據此使開發人員不必擔心加密庫、隨機碼產生或其它加密事務。隨著越來越多加密廠商開始進軍 Secrets 資訊管理市場，我們預計將看到金鑰管理與 Secrets 資訊管理解決方案之間出現大量功能重疊。

- **稽核日誌：**目前，如果您想向企業銷售資安軟體，最好能向企業提供稽核日誌。越來越多平台現在提供日誌檔，有些甚至提供 syslog 與 (或)JSON 格式。內容與篩選品質在很多情況下仍是個問題；但現在，大多數 Secrets 資訊管理工具目前至少包含日誌功能。然而，並非所有解決方案都能夠在一致且不能被未經授權使用者修改或存取的金庫中保存稽核日誌——對大多數企業來說，這是一項重要的資安要求。

- **代理伺服器 (Proxy) 存取：**特權帳號管理 (PAM) 安全與 Secrets 資訊管理之間的界限正變得日益模糊。此項功能意味著 Secrets 資訊管理服務可以對存取憑證或帳密進行保密處理，同時可以提供權杖 (在 Amazon Web Services 環境中則為角色) 來對請求的實體進行授權。

我們列出可協助讀者解決各種具體應用場景問題的這些功能特性。我們的目標是協助您瞭解可用的功能，以及這些功能如何協助滿足您的需求，同時達到 IT 資安要求。我們還希望協助您瞭解為什麼某些產品以它們獨特的方式工作，並介紹您可以期望從市場上獲得哪些功能。

部署考慮事項

最後讓我們看看選擇 Secrets 資訊管理平台時的維運考慮事項。我們將著重於讓產品能夠支援我們的應用場景的具體功能，而不是進行全面的調查來瞭解產品及每種產品如何運作。核心問題包括這些平台如何部署、如何提供延展性與彈性、以及如何與為其提供 Secrets 資訊的服務互相整合等。要瞭解不同產品之間的區別，您需要瞭解產品的開發目的，因為核心功能與部署模式會受到每種平台的目標用途的嚴重影響。

產品類別

Secrets 資訊管理平台可分為兩大基本類別：通用平台與單一功能平台。通用解決方案可以為多種應用場景提供多種類型的 Secrets 資訊。通用平台可以自動提供任何類型應用程式所需的 Secrets 資訊——從向網頁發送使用者名稱與密碼到發放 API 金鑰，再到動態雲端工作負載。單一功能解決方案（通常稱為「嵌入式或原生」解決方案，因為它們安裝在另一個平台中）一般主要用於單一應用場景。例如，幾種嵌入式解決方案著重於佈建 Docker 容器的 Secrets 資訊，並嵌套到您的協作流程管理器中（如：Swarm、Kubernetes 與 DC/OS）等。

瞭解此項區別非常重要，因為嵌入到容器管理器中的產品可能不適合非容器應用場景。好消息是，很多服務是以這種方式部署的，因此嵌入式工具在很多環境中仍然有用；而且由於嵌入式工具利用現有基礎架構，因此一般可以輕鬆整合與擴展。這些平台一般利用其協作流程管理器或容器環境的具體結構來提供 Secrets 資訊。這些平台一般還會假設 Secrets 資訊的使用方式——例如有些平台可能利用 Kubernetes 的命名空間來強制實施原則或利用 UNIX 命名空間來分發 Secrets 資訊。由於容器是暫時的，因此這些 Secrets 資訊管理器通常適合處理暫時的或「動態的」Secrets 資訊。壞消息是，某些嵌入式工具會假設您的叢集是安全的環境，嵌入式工具可以在環境中以純文字形式安全發送與保存 Secrets 資訊。其它嵌入式工具會對 Secrets 資訊進行全面加密，但可能不支援各種類型的 Secrets 資訊或無法整合非容器化應用程式。

著重於單一應用場景的產品可能是您所需的；但要記住，自動化可能會在開發與 IT 環境中的許多不同環節進行，因此可能有所限制。通用產品一般更加靈活且需要更長時間與更大工作量來進行設定，但可以提供專為容器協作流程或密碼管理而開發的工具通常所不能提供的廣泛功能。

部署模式

單台伺服器（Solitary Servers）

孤立伺服器在著重於提高個人工作效率，在早期工具中很常見。孤立伺服器一般包括中央 Secrets 資訊儲存資料庫與負責管理該資料庫的單一伺服器執行個體。基本上所有功能都透過單一服務進行處理，包括使用者介面、儲存管理、金鑰管理、身份驗證與原則管理。這些工具通常透過命令列介面或 API 使用，而且最適合較少數量的系統。

主從架構

對這種模式的稱法因廠商而不同。Primary/Secondary、Manager/Worker、Master/Slave 與 Service/Agent，這只是描述主要服務（負責管理 Secrets 資訊庫）與用戶端（與呼叫應用程式一起運作）間分層關係的一些稱法。這是最常見的架構。經過加密的 Secrets 資訊保存在一個儲存庫中；通常在一個或多個管理器節點間共用或複製這個儲存庫。每個管理器可以與一個或多個代理程式一起運作，支援對應的服務或應用程式。

透過隨需增加使用新的用戶端與伺服器，這種架構可協助提供延展性與可靠性。這些產品通常將每個元件部署為容器，與所支援的應用程式使用相同的基礎架構。許多嵌入式產品是利用這種模式進行擴展。

評估基於主從架構的解決方案時，重點是要瞭解每種解決方案如何處理高可用性與損毀修復，因為不同解決方案處理各種故障場景的能力與架構複雜性程度各不相同。

我們之前討論過在 Secrets 資訊管理工具與接收方（不管是人還是機器）間共用 Secrets 資訊的方式。而且我們也討論與容器管理與協作流程系統的整合（許多工具就是為此設計的）。現在介紹常用的其它整合點及各整合點的工作方式。請注意，在將 Secrets 資訊管理平台與各種系統互相整合所需要的工作量大小與黏合程式碼數量方面，各解決方案可謂天壤地別。

- **組建伺服器**（Build Server）：軟體發展團隊利用 Jenkins 與 Bamboo 等工具來自動開發與驗證新的程式碼。這些工具通常存取一個或多個儲存庫來獲取更新後的程式碼、自動化腳本與庫來建立新的環境，連線到虛擬或雲端伺服器來執行測試，並在將程式碼發送到另一個儲存庫或容器登錄中之前進行程式碼簽名。每項操作都需要提前取得憑證或帳密才能進行。Secrets 資訊管理可以外掛程式元件或外部服務來整合組建伺服器。

- **IT 自動化**：組建伺服器的自動化建構與強大功能大幅提高開發工作效率，但負責以曲速將程式碼從開發人員桌面移動到生產環境中的是協作流程工具。Chef/Puppet/Ansible 是實現 IT 與開發任務自動化的三大最常用協作流程工具，也是連續整合與連續部署的骨幹。幾乎任何可程式設計的 IT 操作都可以利用這些工具完成，包括大多數 VMware 與透過 API 提供的所有雲端服務功能。與組建伺服器一樣，Secrets 資訊管理一般也作為元件或協調流程工具的附加模組安裝，或作為服務執行。

- **公有雲支援**：公有雲是一個特例。從概念上講，本系列中介紹的所有應用場景都適用於雲端服務。由於公有雲中的每種服務皆透過 API 提供，因此公有雲最適合採用 Secrets 資訊管理工具。對於雲端服務，特殊之處在於如何管理整合：可直接整合雲端的大多數 Secrets 資訊管理工具與雲端原生身份系統或雲端原生金鑰管理系統（或二者）。這帶來進一步的優勢，因為 Secrets 資訊可以在任何地區提供給該地區內受支援的任何服務（利用現有身份）。雲端服務可以全面定義哪些使用者可以存取哪些 Secrets 資訊。然後，透過制定適用於 Secrets 資訊的額外使用原則或包裝到另一加密層中，Secrets 資訊管理就可以增強安全性與合規性。還有一些情況下，客戶不希望實現與雲端服務的全面整合，不願意讓某些 Secrets 資訊與加密金鑰掌握在雲端服務提供者的手中，以避免提供商在收到法庭命令時使自己處於不利之地。實作雲端專用解決方案的另一個缺點是不能靈活地更換雲端服務提供者。

總結

在我們利用雲端服務並更依賴自動化來佈建應用程式與 IT 資源的過程中，我們發現安全的將 Secrets 資訊提供給應用程式與腳本的需求水漲船高。自動化與協作 IT 及應用程式而不需要手動提供憑證或帳密的需求，產生需要 Secrets 資訊管理的需求。開發人員清楚，加密金鑰與 API 憑證以不受保護的方式保存在磁碟中，而首要任務是更快速更輕鬆地交付程式碼。現在我們也應該考慮提高安全性。Secrets 資訊管理工具可以解決此問題，而且適用於需要 Secrets 資訊的各種環境。Secrets 資訊管理工具包括 API，因此可以新增到腳本與自動化服務中，完美地應用到 DevOps 維運模式中。

若有有關此主題的任何問題，或希望討論貴公司的具體處境，請發送電子郵件至 info@securosis.com 或在我們的部落格中發問。

關於分析師

Adrian Lane，**分析師 / 首席技術長**（CTO）

Adrian Lane 是擁有 25 年業界經驗的資深資安策略家。Adrian 擁有 10 多年的高階主管專業經驗並且加盟 Securosis。Adrian 主攻資料庫安全性、安全的應用程式開發與資料安全性。作為廠商社群成員（包括在 Ingres 與 Oracle 任職），Adrian 積累豐富的經驗，而且曾長期在甲方端擔任首席資訊長，在安全解決方案實作方面堅持以業務為中心的觀點。在加盟 Securosis 之前，Adrian 曾擔任資料庫資安公司 IPLocks 的首席技術長、Touchpoint 公司工程副總裁、安全支付與數位權限管理公司 Transactor/Brodia 的首席技術長。Adrian 還在 Dark Reading 上發佈部落格文章，而且是《資訊資安》（Information Security）雜誌的固定投稿人。Lane 先生擁有加州大學伯克利分校電腦科學學士學位，並在史丹福大學取得了作業系統專業研究生學位。

關於 Securosis

Securosis 有限公司是一家獨立調研與分析公司，專門致力於思想領袖、客觀性與透明性研究。我們的分析師都曾擔任過企業高階主管職位，致力於提供高價值而且實用的諮詢服務。我們的服務包括：

- Securosis Nexus：Securosis Nexus 是一種線上環境，可以協助您更好更快完成工作。Securosis Nexus 提供有關資安主題的實用研究，有助於您瞭解需要掌握的情況，還提供業界領先的專家建議來回答您的問題。Nexus 的設計快速而且簡便易用，能夠以最快的速度為您提供所需的資訊。請瀏覽：<https:// nexus.securosis.com/>。

- **主要調研成果發佈：**我們目前透過部落格免費發佈大部分調研成果，並在 Research Library 中進行歸檔。大多數這些調研成果可以應贊助商的要求每年發佈。所有發佈的材料與示範符合我們嚴格的客觀性要求以及我們的全面透明研究 (Totally Transparent Research) 政策。

- **適合終端使用者的研究產品與策略諮詢服務：**Securosis 將推出一系列研究產品與基於諮詢的預定服務，有助於終端使用者加速專案實作並取得計畫成功。此外我們還可以提供更多諮詢專案，包括產品選擇輔助、技術與架構策略、教育、資安管理評估及風險評估。

- **適合廠商的預付費 (Retainer) 服務：**儘管我們會接受來自任何人的簡介資料，但某些廠商傾向於建立更緊密的長期合作關係。我們提供多種靈活的套裝預付費服務。預付費服務中包含的服務有市場與產品分析與策略、技術指導、產品評估及併購評估。即使對於付費客戶，我們也會執行嚴格的客觀性與保密性要求。我們可提供有關預付費服務的更多資訊 (PDF)。

- **外部發言與社論：**Securosis 分析師經常在業界活動上發言，進行線上示範，並為各種出版物與媒體撰稿與 / 或發表演講。

- **其它專家服務：**Securosis 分析師還可以提供其它服務，包括策略諮詢日 (Strategic Advisory Days)、策略諮詢服務以及投資者服務。這些服務可以根據客戶的具體要求進行靈活定制。

我們的客戶既有悄然興起的新型企業，又有全球最知名的技術廠商與終端使用者，包括大型金融機構、機構投資者、中型企業及知名資安服務提供者。

此外，Securosis 還與資安測試實驗室合作，提供獨特的產品評估，將深入技術分析與進階產品、架構與市場分析結合在一起。有關 Securosis 的更多資訊請瀏覽我們的網站：<http://securosis.com/>.