# 前言

> 有位新手嘗試重新開機來修復故障的 *Lisp* 機器。*Knight* 見狀後，嚴厲地說：「如果不了解問題的根源，光是重開機是無法解決問題的。」隨後 *Knight* 將機器關機再重開，機器就恢復正常運作了。
>
> —AI 公案（*https://oreil.ly/0rg4Q*）

在過去半個世紀中，麻省理工學院（MIT）一直是駭客文化的重要基地，尤其是其下的人工智慧實驗室，學習與創造是其核心精神。MIT 駭客發展出獨特的語言和文化，不只創造新詞彙，也形成特有的幽默感。上述模仿禪宗公案的引文是一則 AI 公案，以此啟發讀者領悟，是我最喜歡的公案之一，因為它傳達理解事物運作原理的重要性。文中提到的 *Knight* 是 Tom Knight，MIT 人工智慧實驗室一位備受敬重的程式設計師。

本書主要從安全測試的角度切入，讓讀者深入了解 Kali Linux 的各項功能，希望幫助讀者更透徹地理解這些工具的運作方式和基本原理。Kali Linux 是一個專注於資安的 Linux 發行版本，因此特別受到安全測試或滲透測試專業人員及愛好者的青睞。雖然它也能作為一般用途的 Linux 發行版本，並用於數位鑑識和其他相關領域，但它最初就是為了安全測試而設計的；因此，本書著重於介紹 Kali 內建的工具。有時這些工具不易在其他 Linux 發行版本中取得，雖然可透過原始碼編譯，但若套件已存在於發行版本的套件庫中，安裝過程會更加便利。

# 本書內容簡介

基於從安全測試角度介紹 Kali 的主旨，本書包含以下主題：

### *Kali Linux* 簡介

Linux 擁有豐富的歷史，可追溯至 1960 年代的 Unix。本章介紹 Unix 的相關背景，幫助讀者更清楚了解 Linux 工具的運作原理及使用方法。我們也會探討命令列介面，因為在本書後續內容中會經常使用到，同時也會介紹各種可用的桌面環境，讓讀者擁有舒適的工作環境。若讀者是 Linux 新手，本章將為後續內容打好基礎，避免在深入探討各種工具時茫然不解。

### 網路安全測試基礎

最常見的服務都能在網路上運作，加上連接到網路的系統可能存在漏洞，因此，為加強網路測試能力，我們會介紹網路協定的基本運作原理；在深入安全測試時，了解所使用的協定會有極大優勢。我們也會探討可用於網路堆疊和應用程式壓力測試的工具。

### 偵察

執行安全測試或滲透測試時，偵察目標是基本步驟。有多種開放資源可以協助蒐集目標資訊，這不只有助於測試的後續階段，也能為受測組織提供詳細的資訊，並以此協助他們準確評估外界可見的系統範圍。畢竟，攻擊者可能將組織及其成員的資訊用為攻擊跳板。

### 尋找漏洞

組織遭受攻擊的原因在於漏洞。我們會探討漏洞掃描器，這些工具可以深入分析目標組織存在的技術性（而非人為）漏洞。這將提供後續方向的指引，因為安全測試的目標是為受測組織提供潛在漏洞和風險的洞察。找出漏洞將有助於達成此目標。

### 自動化漏洞利用

雖然 Metasploit 是執行安全測試或滲透測試的基本工具，但還有其他可用的工具。我們會介紹 Metasploit 的基本用法，同時本書也會在其他章節討論可用來利用及發現漏洞的其他工具。

### 深入掌握 *Metasploit*

Metasploit 是功能很豐富的軟體，要熟練運用需要投入很多時間。它提供將近 2,000 個漏洞利用模組和超過 500 個負載，這些組合可以產生數千種與遠端系統互動的可能性。此外，使用者還能開發自己的模組。我們會深入介紹 Metasploit，探討漏洞利用的進階使用方式。

### 無線網路安全測試

現今無線網路已無所不在，不論是手機、平板電腦，還是許多筆記型電腦都透過無線網路連接企業網路。然而，不是每個無線網路都採用最佳設定。Kali Linux 提供多種執行無線網路測試的工具，包括無線網路掃描、框架注入和密碼破解。

### 網頁應用程式測試

大量的商業活動都透過網頁介面進行，此外，許多敏感資訊也透過網頁介面來存取，企業需要密切關注其重要網頁應用程式的漏洞情況。Kali 內建多種協助評估網頁應用程式的工具，我們會探討以代理為基礎的測試及其他自動化測試工具，目標是協助受測組織深入了解這些應用程式的安全狀態。

### 破解密碼

雖然密碼破解非必要性，但有時需要測試遠端系統和本地密碼資料庫的密碼複雜度，以及遠端存取的難度。Kali 提供多種密碼破解程式，包括破解密碼雜湊值如密碼檔案，和以暴力破解方式登入 SSH、VNC 等遠端存取協定。

### 進階技術與概念

Kali 工具庫可用於執行各種廣泛的測試。然而，最終還是需要超越既有技術，開發自己的方法，這可能包括建立自己的漏洞利用模組或開發工具。深入了解漏洞利用的運作方式並開發自己的工具，將為未來發展提供明確方向。我們會介紹擴充 Kali 工具的方法，以及常用指令碼語言的基礎知識。

### 逆向工程與程式分析

因為通常無法取得原始碼，了解程式運作方式可能是漏洞測試的重要環節。此外，也需要分析惡意軟體，反組譯、除錯和反編譯等工具可用於這類工作。

數位鑑識

雖然這個主題並非特別針對安全測試，但了解一些鑑識工具仍有其價值，這也是 Kali Linux 內建的工具類別之一。畢竟，Kali 是一個以資安為導向的發行版本，並不局限於滲透測試或其他安全測試。

報告

雖然不是直接的測試工作，但報告非常重要，因為這關係到專案的收費。Kali 提供多種協助產生報告的工具。我們會介紹測試過程中的記錄技巧及產生報告的策略。

## 本版新增內容

本版新增了數位鑑識章節，介紹這個領域中重要的工具集。除了 Wireshark 等其他章節討論的網路工具外，還包括可用於硬碟鑑識、惡意軟體識別和記憶體擷取的工具。

前版的逆向工程與程式分析內容已擴充成為全新的章節，包含 NSA 開發的 Ghidra 工具，以及其他實用的逆向工程和程式分析工具。

當然也涵蓋 Kali 更新版本提供的新工具，但因為工具數量龐大且不斷更新，且有數百個用於各種資安相關用途的工具套件，所以無法完整涵蓋所有工具。

## 本書受眾

雖然希望本書能為不同經驗程度的讀者提供價值，但主要讀者目標是已具備基礎 Linux 或 Unix 經驗，而想更了解 Kali 的讀者。本書也適合想透過 Kali Linux 工具加強安全測試能力的讀者，若已具備 Linux 經驗，可以跳過第 1 章；已有使用常見工具測試網頁應用程式的經驗，但想進一步精進的讀者，也很適合閱讀本書。

## 道德規範的價值與重要性

道德規範這個議題一再提及，因為其重要性值得多次強調。安全測試必須取得授權，未經授權的行為在大多數地區皆為違法；若未經授權擅自探測遠端系統可能會帶來嚴重後果。在一開始就提到法律責任，就是希望能引起大眾重視。

第三章

# 偵察

在執行滲透測試、道德駭客或安全評估時，通常具有特定的參數，這可能包含完整的目標清單，但實際情況往往並非如此，因此，需要先確定系統和人員等目標對象。為了達到這個目的，必須先偵察（*reconnaissance*）。透過 Kali Linux 所提供的工具，可以蒐集到大量關於目標企業及其員工的資訊。

攻擊的目標不只限於系統和在系統上執行的應用程式，也可以鎖定特定人員。在執行滲透測試或紅隊演練時，不一定會將社交工程攻擊列入要求項目中，但這種可能性是存在的；事實上，在現今環境中，社交工程攻擊已成為最常見的初始入侵管道之一。雖然每年的統計數據都有所變動，但根據 Verizon 和 Mandiant 等機構的統計資料顯示，目前企業資料外洩事件中，有很高的比例是來自於社交工程攻擊。

本章將從遠距離的資訊蒐集開始，這樣可以避免目標察覺到偵察行為。然而，在特定階段中，必須與目標互動，因此我們會逐步接近企業所擁有的系統。最後，討論一個重要的概念：連接埠掃描。這個技術能夠提供許多關於系統和其執行應用程式的細節，但透過其他工具和技術所蒐集到的資訊，將有助於更全面地掌握目標的輪廓。

# 什麼是偵察？

在開始之前，先來定義偵察（*reconnaissance*），以確保對此概念的理解一致。根據韋氏大辭典的解釋，偵察是「為了蒐集資訊而進行的初步調查」，這個定義同時也暗示了其與軍事領域的關係。從資訊安全的角度來看，這種軍事相關的暗示很貼切，因為我們經常使用軍事術語，像是軍備競賽、攻擊、防禦以及偵察等。在這個脈絡下，我們的工作就是蒐集資訊，以協助測試人員，也就是攻擊者或對手更有效率地完成任務。儘管在測試期間可以嘗試各種手法，但測試通常都會有其限制，必須謹慎且有效地運用時間，與其在後期耗費大量時間盲目摸索，不如在一開始就投入時間充分了解目標的特性。

在開始蒐集目標資訊時，最好採取低調的方式，以遠端偵察為主，避免與目標直接接觸；當然，這會依專案性質而有所調整。如果是在自家公司內部執行測試，所有人都知道你的工作內容，或許不需要太過低調。不過，仍然可以善用後續將討論的策略，來評估自家公司在網路上留下的足跡。如果發現公司不經意間在公開管道上洩漏許多敏感資訊。可以運用開源情報工具和相關策略來加強公司防護。

### 作業安全

「禍從口出」（Loose lips sink ships）這句二戰時期的警語，簡要地說明了作業安全（*operations security*，*OPSEC*）的核心概念：任務相關的關鍵資訊都必須保密，因為任何資訊外洩都可能危及整個行動。在軍事任務中，這種保密程度甚至延伸到軍事人員的家屬，舉例來說，如果某位家庭成員透露了他們的親人所派往的特定地點，而這位親人又具備特定專業，外界便能推測出軍事行動的實質內容，這就是所謂的「一加一等於二」。同理，當企業公開過多資訊時，不管對手動機為何，都可能分析出許多關於該企業的資訊。因此，落實作業安全的核心要素，對於防範攻擊者和避免資訊外洩給競爭對手來說，都至關重要。

了解企業最需要防範的攻擊者類型也很重要。你可能特別關注競爭對手竊取智慧財產的風險，也擔心來自組織犯罪集團和國家級組織的全面性攻擊。這些不同的威脅類型可以幫助判斷出最需要保護的資訊，和可以公開的資訊。

良好的實踐作業安全，可以有效防範本章即將討論的各種偵察技術。

如果只從網路攻擊的角度考量，執行連接埠掃描和服務掃描或許已經足夠。然而，完整的安全測試並不只有「從開啟連接埠入侵系統」這類強硬攻擊手法，測試範圍可能還包含應變機制、人類介面、社交工程等多個面向；畢竟，一個企業的安全態勢不是只取決於其對外開啟的服務。因此，在準備安全測試時的偵察工作，其實遠不止執行連接埠掃描這麼簡單。

網際網路的一大特性是它儲存了大量資訊。使用者在網路上互動和連結的時間越長，留下的數位足跡就越多，這個現象對個人和企業都適用。以社群網站為例：試想你在上面留下了什麼樣的足跡？在網路上散布了多少個人資訊？作為公司員工又留下了哪些相關資訊？此外，為了維持網際網路正常運作和確保瀏覽更加順暢，系統還儲存了包含域名、聯絡方式、公司詳細資料、地址等，對安全測試來說都很有價值的資料。

隨著時代演進，為了更容易取得這些儲存的資訊，開發者推出許多工具，包含已經運作一段時間的命令列工具，以及網站、瀏覽器擴充功能和其他應用程式。隨著上網人口的增加，資訊蒐集管道也越來越多元，可供挖掘的資訊來源很豐富。雖然可用網站眾多，但以下不會詳細介紹不同網站蒐集資訊的方法，而是著重於 Kali 所提供的工具，並簡單介紹 Kali 預設瀏覽器 Firefox 可安裝的擴充功能。

## 公開來源情報

就在不久前，要找到一個在網路上留下明顯足跡的人，比要找到一個完全不熟悉網際網路的人更困難，但這種情況在短時間內徹底改變了。即使是刻意避開 TikTok、Facebook、X（Twitter）、Instagram 等社群網站的使用者，在網路上仍然會留下足跡。從公開的網路紀錄即可開始追蹤，且任何曾經擁有住家電話的人都能在網路上找出來，包括平時很少使用網際網路的人。至於長期活躍於網路的使用者，留下的足跡更為廣泛，以個人為例，我的網路足跡已經累積了數十年之久。

什麼是公開來源情報（*open source intelligence*）？簡單來說，任何從公開管道取得的資訊都可以算是公開來源情報，不論是歸類為公開的政府紀錄，例如不動產交易資料，還是其他公開來源，如可視為開放資訊來源的郵件群組存檔。聽到 *open source*（公開來源）這個詞，可能會馬上聯想到軟體，但這個概念同樣適用於其他資訊領域。公開來源只表示這些資訊來自可以自由取得的管道，因此，那些需要付費才能查詢個人詳細資料的網站不屬於這個範疇。

為什麼要使用這些公開來源情報呢？這並非為了跟蹤個人，而是在執行安全測試時有多重用途。首要目的是蒐集關於 IP 位址和主機名稱的詳細資訊，假設你需要以完整的紅隊模式測試某家企業，也就是說以外部人員的身分，在沒有獲得任何目標資訊的情況下完成約定目標，這時必須了解攻擊對象，這代表需要找出可以攻擊的系統，也需要識別企業內的員工，因為他們往往是最容易突破的入侵管道。社交工程通常是取得存取權限很有效的方式。

如果你是企業的資安專業人員，需要追蹤公司和高階主管在外部留下的足跡。企業可以減少對外洩露的資訊來降低遭受攻擊的風險，當然，這些資訊不可能完全消除，至少，關於域名、指派給企業的 IP 位址以及 DNS 紀錄的資訊必然存在。如果這些資訊不是公開的，消費者和供應商、合作夥伴等其他企業，就無法與之互動。

搜尋引擎能提供大量資訊，是個很好的起點，但網際網路上的網站數量龐大，很容易會被大量搜尋結果淹沒，其中一個解決方案是縮小搜尋範圍。雖然這與 Kali 沒有直接關係，而且許多人都知道這個技巧，但仍是個值得快速回顧的重要主題。安全測試時，需要大量資訊搜尋，善用搜尋技巧可以省下閱讀無關頁面的許多時間。

執行社交工程攻擊時，需要確定攻擊對象，這表示要識別目標企業的員工。社群網站可以用來蒐集大量關於個人的資訊，LinkedIn 是識別企業及其員工的重要資料來源之一，求職網站也能提供許多關於企業的資訊。舉例來說，如果發現某家企業正在徵求具備 Cisco 和 Microsoft Active Directory 經驗的人才，就能推測該企業使用的基礎架構類型。其他社群網路如 LinkedIn 和 Facebook 也能提供一些關於企業和人員的資訊。

這確實需要搜尋大量資訊，幸好，Kali 提供了尋找這些資訊的工具，這些程式可以自動從搜尋引擎和其他網路位置擷取大量資訊。其中 theHarvester 這類工具不只操作簡單，還能節省大量時間；Maltego 這類程式則不只能自動擷取大量資訊，還能以視覺化的方式呈現資訊之間的關係。不過，在介紹各種工具之前，我們應該先了解一個基本且高效率的資訊蒐集方法。

## Google 駭客技術

在 Google 出現之前，搜尋引擎就已經存在了，然而，Google 改變了搜尋的運作方式，因此超越了當時的主流搜尋網站如 AltaVista、Infoseek 和 Inktomi 等，這些網站後來都被收購或停止營運，許多其他搜尋引擎也已經停止服務。Google 不只建立了一個實用的搜尋引擎，還找到將搜尋引擎商業化的方法，使公司能夠維持獲利並持續營運。

Google 採用的其中一個功能是一組關鍵字，使用者可以運用這些關鍵字來調整搜尋請求，以獲得更精確的搜尋結果。而這樣使用關鍵字的搜尋方式有時會稱為 *Google 呆瓜搜尋*（*Google dorks*），使用關鍵字來識別特定頁面的整個過程則稱為 *Google 駭客技術*（*Google Hacking*）。當嘗試蒐集目標資訊時，這些技術可以成為一組極為強大的工具。

在篩選特定目標相關資訊時，最重要的關鍵字之一是 *site:*，使用這個關鍵字時，能限制 Google 只搜尋符合特定網站或域名的結果，例如使用 *site:oreilly.com*，表示只搜尋所有以 *oreilly.com* 結尾的網站頁面，包括 *blogs.oreilly.com* 或 *www.oreilly.com* 等這類網站。此功能實際上讓每個組織具備如同內建 Google 搜尋引擎的網站架構功能，並且能使用 Google 在相同網域下執行跨站搜尋。

雖然組織擁有類似搜尋引擎的功能，但使用這種技術時，需注意只能搜尋到可從網際網路存取的頁面和網站。即使網站可從網際網路存取，若無相關連結指向該網站，搜尋引擎也無法發現它們：這代表無法搜尋任何內部網路網站或頁面，因為這類網站通常只能在組織內部存取。搜尋引擎爬蟲必須知道網站存在才能爬取，因此若是設定錯誤時，有人從公司外部連結到可從外部存取的內部網站，就可能導致該網站曝露於搜尋引擎的爬蟲程式中。但一般來說，無法從外部取得內部網站的詳細資訊。

你可能會想要限制特定的檔案類型，例如，尋找試算表或 PDF 文件，這時可以使用 *filetype:* 關鍵字，將搜尋結果限制在特定檔案類型。實際上，可以同時使用多個關鍵字來獲得更精確的結果，圖 3-1 中，搜尋字串為 *site:oreilly.com filetype:pdf*，這會顯示 Google 在所有以 *oreilly.com* 結尾網站上找到的 PDF 文件，可以看到前兩個搜尋結果就來自不同的網站。

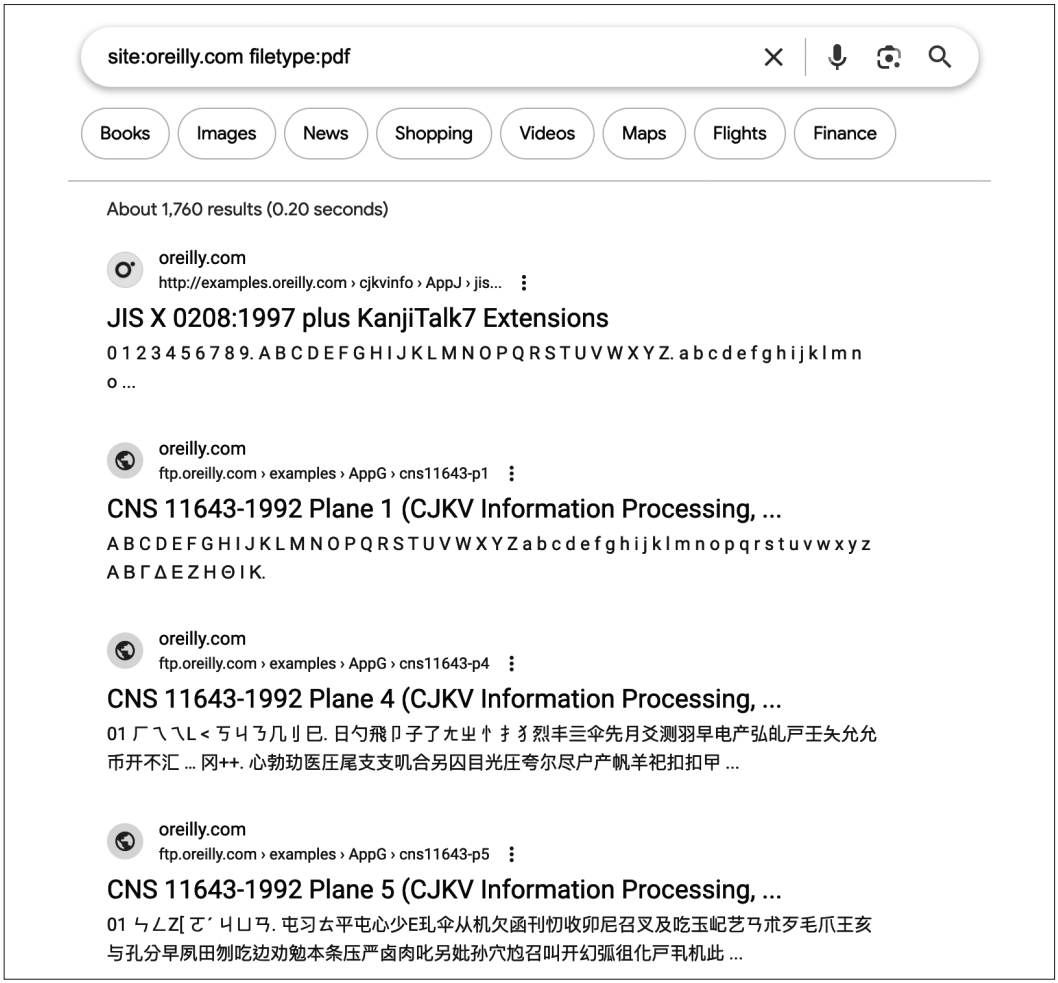


圖 3-1　Google 的 filetype 和 site 搜尋結果

另外還有兩個值得注意的關鍵字：*inurl:* 和 *intext:*，前者會在 URL 中尋找搜尋字，後者則是在網頁內文中尋找搜尋字。一般情況下，Google 會在頁面的相關元素中尋找符合的內容，而使用這些關鍵字則是明確告訴 Google 要限制搜尋字的範圍，如果要尋找 URL 中包含類似 */cgi_bin/* 的頁面，這些關鍵字特別有用。也可以使用 *intext:* 後面加上搜尋字，指定 Google 只在頁面的文字內容中尋找符合的文字。一般情況下，Google 會顯示不包含所有搜尋字的結果，如果想確保找到所有內容，可以使用相似的關鍵字 *allinurl:* 和 *allintext:*。

還有其他各種關鍵字，並會隨時間更新，例如已經停用的 *link:* 關鍵字，而上述所提都是常用的主要關鍵字。請留意，通常可以同時使用多個關鍵字，也可以使用基本的搜尋運算，包括布林運算子。例如，可以使用 AND 或 OR 來告訴 Google 想要包含正在搜尋的兩個字（AND），或其中任一個字（OR），也可以使用引號來確保得到正確順序的文字組合，例如，想搜尋自由女神像的相關資料，可使用「Statue of Liberty」這個專有名詞，否則會一次得到包含 *statue* 和 *liberty* 這兩個字的所有頁面，搜尋結果可能包含大量無關內容。

另一個值得注意的 Google 搜尋資源是實用的搜尋查詢資料庫，這是 Johnny Long 在 2004 年建立的 Google 駭客資料庫，他從 2002 年就開始蒐集實用或有趣的搜尋字。目前，Google 駭客資料庫（*https://oreil.ly/oRO3k*）託管在 *exploit-db.com*，這些呆瓜搜尋按類別整理，包含許多有趣的關鍵字，為公司執行安全測試時可使用許多實用關鍵字。可以在資料庫中找到任何搜尋字，再加上 *site:* 和域名，就可以使用 Google 駭客技術找出潛在的易受攻擊頁面和敏感資訊。

最後一個可用關鍵字是 *cache:*，但使用時機較為有限，它可以從 Google 的搜尋快取中擷取頁面，查看 Google 上次快取時頁面的樣貌。就快取結果而言，由於無法指定特定日期範圍，這個關鍵字可能不如 Wayback Machine（*https://oreil.ly/HT3oY*）實用。不過，如果某個網站因為不管什麼原因而無法存取時，可以藉此從 Google 下載該頁面。請留意，若使用 Google 快取瀏覽無法存取的網站時，頁面中的連結仍會指向原始網站，因無法存取而無法點選，需要再次使用 *cache:* 關鍵字來取得該頁面。*cache:* 關鍵字會呈現最新的網站快取版本，搜尋結果會顯示該快取副本的儲存時間；我以 O'Reilly 的網站進行時，收到的是幾個小時前的副本。有些網站的快取頻率可能比其他網站低，雖然可以使用這個關鍵字來參考網站，但搭配完整 URL 可能更為實用，如果有一個指向特定頁面的 URL，可以使用該 URL 從 Google 快取中取得該頁面內容。

## 自動化資訊擷取

這些搜尋作業很耗時，尤其是需要執行多個查詢以獲得最完整的結果時，還好，我們可以運用 Kali 中的工具來快速取得結果。首先要介紹的工具是 *theHarvester*，一個能夠從多個來源蒐集詳細資訊的程式，它不只支援百度（Baidu）、Bing 和 DuckDuckGo 等主流搜尋引擎，還包含一些安全導向的搜尋網站，例如用於威脅情報的 ThreatMiner，以及用於搜尋 DNS 資料的 DNSDumpster。

在過去，theHarvester 可用於蒐集關於人員的資訊，因為它能夠搜尋用於加密電子郵件和其他資料的優良保密協定（PGP）金鑰；此外，theHarvester 也可以搜尋 LinkedIn。雖然搜尋網域相關電子郵件地址的功能已大幅降低，但在識別網域相關的 IP 位址和完整網域名稱（FQDN）方面仍然十分有效。這個工具在尋找電子郵件地址方面仍然可行，因為從輸出結果可以看出它會搜尋這類資訊，但在搜尋 O'Reilly 的網域時卻沒有找到任何電子郵件地址。

範例 3-1 使用 theHarvester 搭配 sitedossier 資料來源，搜尋 *oreilly.com* 的 FQDN。不同資料來源會產生不同結果，某些公開來源情報會提供大量與 *oreilly.com* 相關的 FQDN 清單。這次搜尋回傳了 200 筆結果，如果自行執行，可能會得到不同的結果，因為公開系統和其相關的 IP 位址並非靜態的，搜尋結果已經編輯過，以便了解各個部分的呈現方式。

*範例 3-1　theHarvester 的 sitedossier 結果*

```
┌──(kilroy@badmilo)-[~]
└─$ theHarvester -b sitedossier -d oreilly.com
Read proxies.yaml from /home/kilroy/.theHarvester/proxies.yaml
*******************************************************************
*  _   _                                             _             *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.6.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: oreilly.com

My current iter_url: http://www.sitedossier.com/parentdomain/oreilly.com/101
My current iter_url: http://www.sitedossier.com/parentdomain/oreilly.com/201
In total found: 200
{'security.oreilly.com', 'ofps3.vz.oreilly.com', 'chimera.labs.oreilly.com', ↩
'corporate.oreilly.com', 's.radar.oreilly.com', 'apache.oreilly.com', ↩
'access.safari.oreilly.com', 'opengovernment.labs.oreilly.com', ↩
'register.oreilly.com', 'ajax.oreilly.com', 'portal.oreilly.com', ↩
'beautifulcode.wiki.oreilly.com', 'libraries.oreilly.com', ↩
'ormstore-staging.oreilly.com', 'programming-scala.labs.oreilly.com', ↩
'm.bookworm.oreilly.com', 'news.oreilly.com', 'hacks.oreilly.com.', ↩
```

```
'macruby.labs.oreilly.com', 'nutshells.oreilly.com', 'etel.wiki.oreilly.com', ↩
'mediaservice.oreilly.com', 'stats.oreilly.com', 'cachefly.oreilly.com', ↩
'scifoo13.wiki.oreilly.com', 'agiledev.97things.oreilly.com', ↩
'conference.oreilly.com', 'community.toc.oreilly.com', ↩
'dev-blogs.oreilly.com', 'ignite.oreilly.com', 'bio.oreilly.com', ↩
'rails-nutshell.labs.oreilly.com',
<snip>
[*] Searching Sitedossier.

[*] No IPs found.

[*] No emails found.

[*] Hosts found: 199
---------------------
97things.oreilly.com
academic.oreilly.com
access.safari.oreilly.com
actionscript.oreilly.com
admin.members.oreilly.com
agiledev.97things.oreilly.com
ajax.oreilly.com
akamaicovers.oreilly.com
amazon.oreilly.com
androidcookbook.oreilly.com
animals.oreilly.com
annoyances.oreilly.com
answers.oreilly.com
answers.oreilly.com.
answersstage.oreilly.com
apache.oreilly.com
apprenticeship-patterns.labs.oreilly.com
apprenticeship.oreilly.com
architect.97things.oreilly.com
assets.en.oreilly.com
assets.oreilly.com
atom.oreilly.com
beautifulcode.wiki.oreilly.com
bio.oreilly.com
blogs.oreilly.com
```

範例 3-2 為使用 DuckDuckGo 搜尋引擎的結果，相較於 sitedossier 所提供的清單數量較少。雖然針對偵察和情報蒐集的網站能提供數十筆結果，DuckDuckGo 聲稱找到 9 個主機，但實際上只列出其中 5 個。

範例 *3-2*　*theHarvester* 的 *DuckDuckGo* 結果

```
┌──(kilroy@badmilo)-[~]
└─$ theHarvester -d oreilly.com -b duckduckgo
*******************************************************************
*  _   _                                             _            *
* | |_| |__   ___    /\  /\__ _ _ ____   _____  ___| |_ ___ _ __  *
* | __|  _ \ / _ \  / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__| *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |    *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|    *
*                                                                 *
* theHarvester 4.3.0                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************

[*] Target: oreilly.com

[*] Searching Duckduckgo.

[*] No IPs found.
[*] No emails found.
[*] Hosts found: 9
 ---------------------
conferences.oreilly.com
learning.oreilly.com
oreilly.com
radar.oreilly.com
toc.oreilly.com
```

theHarvester 的每個來源都採用不同技術來蒐集資訊，因此同時搜尋多個來源能有效取得資料。範例 3-3 是一個簡單的 Python 指令碼，它會根據在命令列中提供的網域名稱，依序執行每個提供者的搜尋，若有多位使用者需要使用此指令碼，且不熟悉其運作方式，建議增加更多功能。不過就個人使用而言，目前的功能已算實用了，執行結果將產生多個 XML 及 HTML 檔案，其中包含各個提供者所回傳的搜尋結果。

範例 *3-3*　使用 *theHarvester* 搜尋的指令碼

```
#!/usr/bin/python

import sys
import os

if len(sys.argv) < 2:
```

## Maltego

作為一位從無圖形使用者介面時代走過來的人，我比較偏好命令列介面；當然，Kali 確實提供了許多命令列工具，不過，也有些使用者偏好的是圖形介面。之前探討過許多可從開放來源取得大量資料的工具，但這些工具存在兩個明顯的局限：一是難以直觀地看出各項資訊之間的關係，二是缺乏從現有資料延伸取得更多資訊的便捷方式。雖然可以匯出從 theHarvester 或 Recon-ng 獲得的聯絡人清單，再將其輸入到另一個模組或工具中，但若能直接選取某筆資訊並對其執行其他模組，效率會更高。

這正是 *Maltego* 的價值。這款圖形介面程式不只能完成先前執行過的工作，更重要的是，它能以視覺化方式呈現資料，讓所有實體之間的關係一目了然，且獲得一系列實體後，還能從這些實體延伸取得額外資訊。這種循環式的資訊探索過程，能不斷引導而挖掘出更深層的資訊。

在深入了解 Maltego 之前，需要先釐清一些基本術語。Maltego 使用轉換器（*transform*）來執行工作，這是一段使用 Maltego 指令碼語言（MSL）撰寫的程式碼，它能利用資料來源將一個實體轉換成另一個實體。舉例來說，當有一個主機名稱實體時，可以套用轉換器來建立一個新實體，其中包含與該主機名稱相關聯的 IP 位址。如前所述，Maltego 以圖形方式呈現資訊，每個實體都作為圖形中的一個節點。

雖然 Paterva 也提供商業版本，但這裡會使用 Kali 內建的 Maltego 社群版，具有可限制的可安裝轉換器數量，商業版則提供來自不同來源的更多轉換器。儘管如此，社群版仍然提供很豐富的轉換器，可以在圖 3-2 中看到轉換器套件的清單。

| | | | | | |
|---|---|---|---|---|---|
| Fraud-check IP address [IPQS] | Ready | Maltego IPQu... | <none> | IPv4 Address [maltego.IP... | IPv4 Address [maltego.IP... |
| Get tags and indicators (VPN, Tor, Proxy, et | Ready | Maltego IPQu... | <none> | IPv4 Address [maltego.IP... | IPv4 Address [maltego.IP... |
| Get tags and indicators for email address [ | Ready | Maltego IPQu... | <none> | Email Address [maltego.... | Email Address [maltego.... |
| Get tags and indicators for phone number [ | Ready | Maltego IPQu... | <none> | Phone Number [maltego.... | Phone Number [maltego.... |
| Lookup phone number [OpenCNAM] | Ready | Maltego Ope... | <none> | Phone Number [maltego.... | Person [maltego.Person] |
| Mirror: Email addresses found | Ready | Maltego CTA... | <none> | Website [maltego.Website] | Email Address [maltego.... |
| Mirror: External links found | Ready | Maltego CTA... | <none> | Website [maltego.Website] | Phrase [maltego.Phrase] |
| Parse meta information | Ready | Maltego CTA... | <none> | Document [maltego.Docu... | Person [maltego.Person]... |
| Search Page Titles [Wikipedia EN] | Ready | Maltego Wiki... | <none> | Phrase [maltego.Phrase] | Page [maltego.wikimedia... |
| Search Text In Pages [Wikipedia EN] | Ready | Maltego Wiki... | <none> | Phrase [maltego.Phrase] | Page [maltego.wikimedia... |
| To AS Number [WhoisXML] | Ready | Maltego Whoi... | <none> | Netblock [maltego.Netblo... | AS [maltego.AS] |
| To AS [WhoisXML] | Ready | Maltego Whoi... | <none> | Netblock CIDR [maltego.C... | AS [maltego.AS] |
| To Aliases [mentioned in Tweet] | Ready | Maltego CTA... | <none> | Tweet [maltego.Twit] | Alias [maltego.Alias] |
| To CPEs [Shodan Internet DB] | Ready | Shodan Inter... | <none> | IPv4 Address [maltego.IP... | Phrase [maltego.Phrase] |
| To CVE [Shodan Internet DB] | Ready | Shodan Inter... | <none> | IPv4 Address [maltego.IP... | CVE [maltego.CVE] |

圖 3-2　Maltego 社群版可用的轉換器

Maltego 的核心功能在於其內建的轉換器。不過，不需要逐一手動套用轉換器來執行所有工作，這些操作可以交由機器（*machine*）自動完成，它能從指定的起始點開始依序執行轉換器。以企業資訊足跡分析為例，可以使用整合了 DNS 查詢與系統關聯性分析等功能的轉換器，來執行這項工作。具體來說，Footprint L3 機器會根據輸入的網域，自動執行轉換器來擷取郵件交換器（MX）和名稱伺服器（NS）等紀錄。接著，它會從主機名稱解析出 IP 位址，並進一步探索，找出相關聯的主機名稱和 IP 位址。要啟動機器功能，只需點「Run Machine」按鈕，選擇要使用的機器，並輸入所需的參數即可。圖 3-3 可以看到執行機器的對話視窗，上方為含有「Run Machine」按鈕的「Machines」分頁

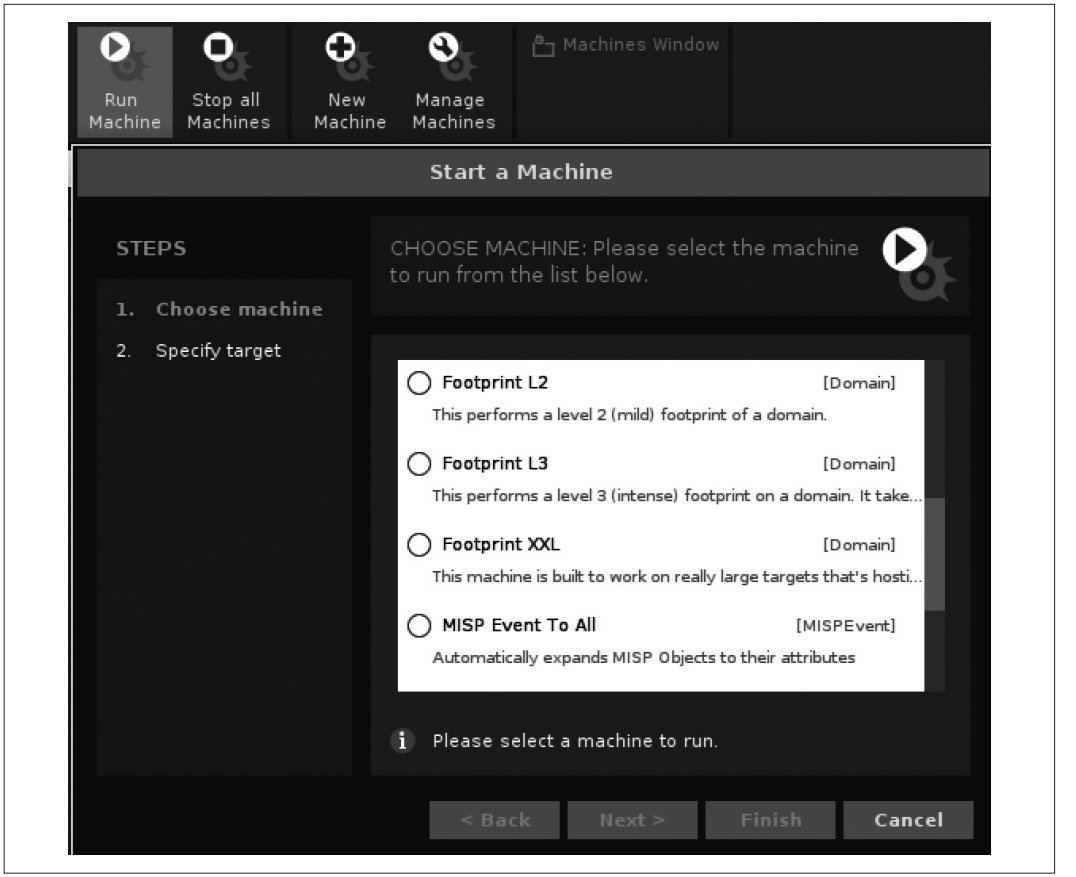


圖 3-3　從 Maltego 執行機器

在執行過程中，機器會詢問偏好設定，以確認要納入或排除的實體；待機器運作完成後，將獲得一張關聯圖，與一般常見圖表不同，這張圖是用來呈現實體間關係的有向圖。在機器產生的圖表中央，會看到最初輸入的網域名稱，從中心點往外延伸的則是各種不同類型的實體，每種實體都以獨特的圖示表示類型。舉例來說，類似網路介面卡外觀的圖示代表 IP 位址實體；而系統堆疊狀的圖示則可能代表 DNS 或 MX 紀錄，這些可以透過不同的顏色來區分。圖 3-4 是一個實際的 Maltego 關聯圖範例。

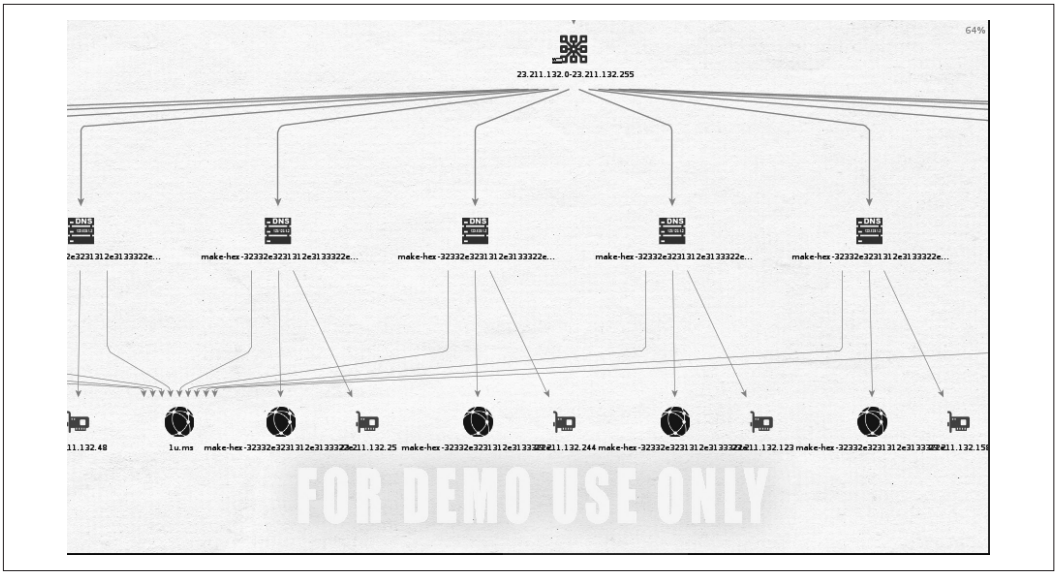


圖 3-4　Maltego 中的有向圖

在每個實體上按右鍵，可以呼叫出環境選單，查看能套用於該實體的所有轉換器。例如，知道主機名稱但缺少 IP 位址時，就能使用轉換器來查詢相關 IP。如圖 3-5 所示，可從區域網際網路註冊機構取得與該實體的相關資訊，此為 ThreatMiner 提供的 whois 轉換器。

每執行一次轉換器，就會擴大圖表規模；擁有的轉換器越多，能蒐集的資料就越豐富。從單一實體開始，便能快速取得大量資訊，這些資訊會以有向圖呈現，能一目了然地掌握實體之間的關係。只需點選任何實體，就能輕鬆查看其詳細資訊，包含相關的來源及目標實體。這種視覺化的呈現方式，不只能清楚展現實體間的連結關係，還能追溯資料的來源。

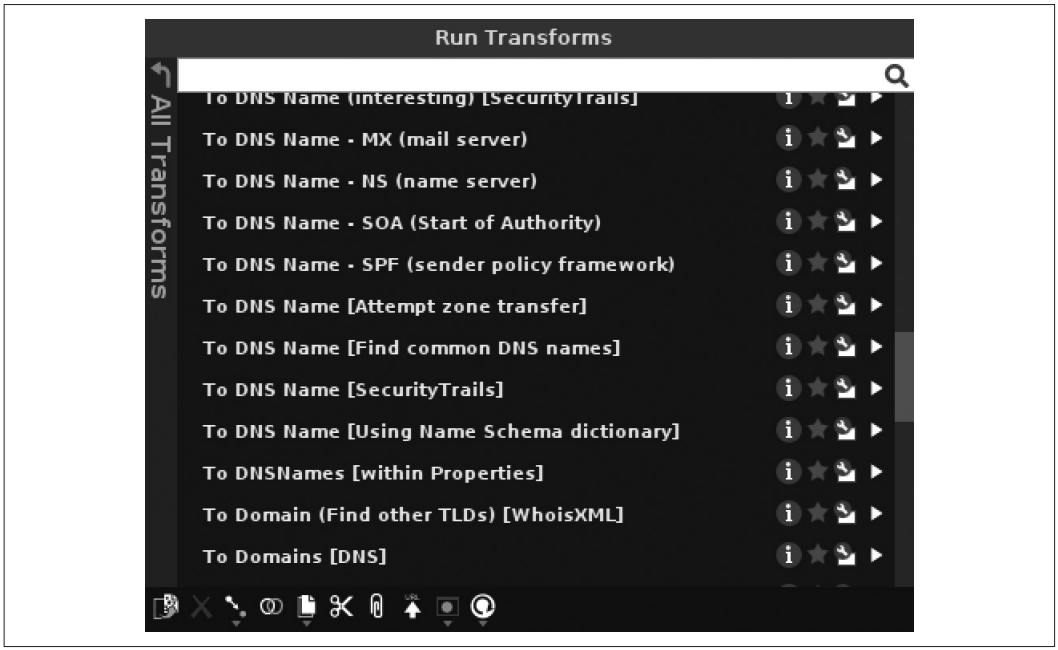


圖 3-5　可套用於實體的轉換器

如果你是視覺化思考者，偏好透過圖像來理解資訊之間的關聯，Maltego 會是一個很好的選擇。當然，也可以透過其他方法來取得相同的資訊，但這些方式會比較費工，而且需要更多的打字時間。

## DNS 偵察與 whois

現今的網際網路本質上是以 DNS 為核心在運作，這也解釋了 DNS 的資安漏洞向來備受關注之因。如果沒有 DNS 系統，我們必須在腦中記住一份龐大的主機對照表，因為所有的網路通訊都得仰賴 IP 位址，包括那些經常變動的位址，這正是 DNS 系統誕生的初衷。在 DNS 出現以前，整個網路只依靠一個主機檔案來維護 IP 位址與主機名稱的對應關係，要知道，那時的主機是大型的多人共用系統，每當網路中增加新的主機時，就必須更新主機檔案，然後分發給所有使用者，這種做法顯然無法因應網路的快速成長。於是，DNS 應運而生，將維護主機名稱與 IP 位址對應關係的任務分散化。