前言

Web 應用程式遍及所有領域與每個產業，從零售業到銀行業，從人力資源到博弈娛樂，其觸角深入 Web 世界裡的每個角落。今日，小如個人的部落格，大到至關重要的金融應用程式，都被建構成某種形式的 Web 應用程式。如果我們打算將應用程式成功地移植到 Web 上，或者在 Web 上建造新的應用程式，就必須能夠有效地測試那些應用程式。誠然，單單功能測試就足以應付客戶所需的日子已不復存在，今天，Web 應用程式正面臨著無所不在且不斷增長的安全威脅，駭客、內部人士、罪犯們正虎視眈眈地覬覦著你的心血結晶。

這本書全然關乎我們如何測試 Web 應用程式，特別是有關安全測試的面向，「我們」指的是開發者，測試者，架構師，品管經理、以及有需要測試 Web 軟體的顧問。不管遵循什麼樣的品質或開發方法，將安全測試納入我們的測試涵蓋範圍中，必須用到新的測試模式。此外，我們也需要用到專業化的工具，讓安全測試的相關工作變得更容易履行。貫穿本書所有的錦囊妙技（秘技），我們會善用 Web 應用程式的同質性（homogenous nature），盡可能利用一些有關 Web 應用程式放諸四海皆準（或通常成立）的事實，這樣的共通性會讓本書的秘技適用於多方用途，更可能為你所利用，此外，這也表示你將發展出多用途的測試工具，而不是針對單一應用程式。

誰適合讀這本書

本書係針對主流開發者與測試者所撰寫的，而不是安全防護專家。任何涉及 Web 應用程式開發的人應該都能夠在這裡找到一些有價值的東西。負責為元件撰寫單元測試（unit test）的開發者，會喜歡這些工具能夠確切聚焦在單一頁面、功能、或表單的模式；負責測試整個 Web 應用程式的 QA 工程師，會對測試案例的開發與自動化特別感興趣，這些測試案例很容易就能夠成為回歸測試組（regression test suite）的一部分。本書的秘技主要運用免費工具，不需要你投入一分一毫的金錢，只要憑藉著個人的努力，就能夠將這一切應用在實際的工作上。

本書秘技所選用的工具與處理的任務都是與平台無關的，這表明兩件非常重要的事情：它們可以在你的桌上型電腦裡頭執行，不管它跑的是什麼樣的作業系統（Windows、MacOS、Linux、等等），並且可以跟你的 Web 應用程式合作無間，不管它是透過什麼樣的技術而建造的，Java、PHP、ASP、CGI、以及任何其他的 Web 技術都一樣。在一些案例中，我們會宣稱某個任務係特定於某種環境，但那通常是額外的補充，而不是該祕技的主要焦點。因此，本書的讀者可以是在任何 Web 平台上運用任何技術的開發者或測試者。你不需要用到什麼特別的工具（除了本書所討論的免費工具），或者能夠運用這些技術的特殊環境。

運用免費工具

有很多免費的測試工具能夠用來幫助開發者或測試者為應用程式的基本功能進行安全測試，這些工具不僅免費，而且非常容易客製化，非常有彈性。在安全測試的領域中（或許超過任何其他的 QA 專業領域）最好的工具傾向是免費的。即使在網路安全領域中，現今的商用工具可謂既成熟且威力強大，但是，在它們能夠跟有用的免費工具相互競爭之前，也著實花費了一段好長的時間，縱然是現在，也沒有哪個網路安全防護人員僅僅使用商用工具來從事他的工作，免費工具仍然扮演著極關鍵的角色。

不過，在許多情況下，免費工具往往缺乏文件說明，那正是本書所填補的隙縫之一：告訴你如何充分運用一些你可能聽過的，但缺乏良好文件說明（描述如何及為何使用它們）的工具。我們認為某些主流開發者與測試者就是因為不知道如何下手，因而失去從這些免費且容易取得之工具獲得利益的好機會。

另一個運用免費工具有效測試 Web 應用程式的障礙是：普遍缺乏如何將這些免費工具組織起來以執行良好安全測試的知識。知道 TamperData 能夠讓你規避掉客戶端檢查是一回事，利用 TamperData 發展出良好的跨網站指令稿測試又是另一回事。我們想要讓你超越「進行良好的 Web 應用程式測試」的層次，進而產生良好的安全測試案例，並且從中獲得可靠的結果。

最後，因為許多開發與 QA 團隊都沒有充足的工具與訓練預算，側重於免費工具代表你能夠毫不受限地嘗試這些祕技，而不需要取得某項昂貴工具的展示版授權。

關於封面

封面上的鳥被稱為「星鳥」(nutcracker [譯註]，又叫作 Nucifraga Columbiana)，很適合作為 Web 安全測試的吉祥物。星鳥試圖打開未成熟的松果，取出裡頭的種子，它們的

譯註　nutcracker 也是胡桃鉗的意思。

鳥喙被設計來深入探查隱匿的裂縫，將食物取出。同樣的，身為安全測試人員，我們試圖利用專業工具，打開應用程式，存取裡頭的私有資料、需要特殊權限的功能、以及開發者不願看到的操作行為。這本書所扮演的角色之一，就是提供你許多可以深入運用的專業工具，另一個角色則是提示你臭蟲躲藏的隱匿裂縫究竟會在哪裡。

星鳥還有一項非常出色的天賦，就是能夠記住並且回到它先前用來藏匿食物的所有地點，它將收集到的種子存放在成百上千個貯存處所，然後利用它們來度過整個寒冬。我們的測試活動再一次跟星鳥的行為相仿，因為我們也會建構一系列的回歸測試，記錄我們在應用程式當中所找到的安全漏洞。理想上，運用本書的工具與技術，我們會重新檢視先前所找到的問題，確認那些問題已經被解決，並且持續保持不再出現。

更多有關星鳥的資訊，請參考康乃爾大學的 The Birds of North America Online 網頁，*http://bna.birds.cornell.edu/bna/*；更多有關 Web 安全測試的資訊，請繼續閱讀本書。

本書組織

本書主要分成三個部分。第一個部分涵蓋工具的設定以及一些用來發展測試的基礎概念。第二個部分聚焦在針對不同目的規避客戶端輸入驗證的各種方法（SQL 注入、跨網站指令稿、竄改表單隱藏欄位、等等）。最後一個部分把重點放在期程（session），找出期程識別符、分析它們的可預測性、以及運用工具操作它們等。

每個祕技的闡述皆遵循共同的格式，從要解決的問題開始，繼而說明要用到的工具與技術、測試程序，最後是範例與討論。所有的祕技皆是以測試的角度出發，也就是說，你會對某個祕技有興趣，因為它會讓你更容易測試到 Web 應用程式的某個安全面向。

本書的組織從基本的任務展開，逐步擴展到高階的任務。每個主要的部分都是從相對簡單的任務開始，再逐漸深入更複雜的任務。前面的祕技純粹只是一些簡明清晰的練習，說明 Web 應用程式背後所發生的事情；後面的祕技會將許多建構區塊組織成複雜的任務，形成大型 Web 應用程式安全測試的堅實基礎。

第一部分：基礎

我們從建置測試環境開始，這個部分幫助你熟悉貫穿本書所需的基礎知識。你需要學習的第一件事情是瞭解如何安裝、設定、及操作工具，接著，你必須明白我們會加以利用的 Web 應用程式之共通特性，好讓我們的測試盡可能適用在各種情境之中。

第 1 章，簡介，將我們對軟體安全測試以及如何將它運用到 Web 應用程式的想法，完整地呈現給你，其中也包括一些本書會參照到的術語及重要測試概念。

第 2 章，安裝免費工具，涵蓋完整的工具組，包括你能夠下載及安裝的各種免費工具。每一項工具皆包含一些有關在哪裡可以找到它，以及如何安裝與執行的基本指示。稍後，我們會利用這些工具，進行實際的安全測試。

第 3 章，基本觀察，教導你觀察 Web 應用程式並且深入測試系統功能性的基礎知識。你必須運用這些基本技術，才能夠進行本書後面更進階的祕技。

第 4 章，**Web** 導向的資料編碼，說明各種資料編碼，你必須瞭解 Web 應用程式如何以各種不同方式為資料進行編碼（encode）與解碼（decode）。除了編碼與解碼之外，你必須能夠以目視的方式判斷編碼過的資料，瞭解它是如何被編碼的。你必須解碼、竄改、再編碼，以便進行一些測試。

第二部分：測試技術

本書第二個部分提供一些基礎的測試技術，為你說明兩種掃描技術：手動掃描（manual-scanning）與大量掃描（bulk-scanning）。這部分的章節涵蓋一般工具與特定工具，讓你進行各種不同的工作，並且將它們組合成更複雜的測試任務。

第 5 章，竄改輸入，討論最重要的基礎技術：惡意的輸入。如何將它運用到你的應用程式？如何看到瀏覽器裡頭所發生的事情？以及它到底傳送了什麼給 Web 應用程式？

第 6 章，自動化大量掃描，介紹幾種進行大量掃描（bulk-scanning）的技術與工具。我們會說明如何以網站搜索器（spider）掃描你的應用程式，找出輸入點與輸入頁面，以及如何針對某些特殊應用程式進行批次測試（batch test）。

第 7 章，使用 **cURL** 自動化特定任務，介紹一種建造自動化測試的好工具：cURL。我們會介紹幾個提交整批測試的常見模式，再逐步發展出更困難的任務，像是在登入與竄改 cookie 時，保留住當時的狀態，最後會建立一項相當複雜的任務：登入 eBay。

第 8 章，**LibWWWPerl** 自動化測試，聚焦在 Perl 和它的 LibWWWPerl 程式庫（LWP）。這一章不是在教導 Perl 編程，而是說明一組能夠搭配 Perl 及 LWP 程式庫進行重要安全測試的特定技術，包括上傳病毒給應用程式、測試長度不合理的檔案名稱、剖析應用程式的回應，以及最精采的部分：能夠編輯 Wikipedia 網頁的指令稿。

第三部分：進階技術

最後幾章的進階技術奠基於本書前面的祕技，我們以各種方式將它們結合起來，以便完成更多測試，或者進一步補充先前祕技未說明清楚的安全測試。

第 9 章，尋找設計瑕疵，討論 Web 應用程式當中一些未預期的互動，並且說明如何以良好的安全測試發掘它們。本章所探討的設計瑕疵包括可預測的識別符、不良的隨機性、以及可重複操作的交易、等等。

第 10 章，攻擊 **Ajax**，說明許多重要的 Web 攻擊，以及如何運用先前所學的技術，以系統化和測試聚焦（test-focused）的方式，執行它們。注入 SSI（Server-Side Includes）、濫用 LDAP、及 SQL 注入都是本章所討論的攻擊。

第 11 章，操作期程，深入檢視 Ajax，一種主導所謂的 Web 2.0 應用程式的技術。我們會說明如何深入 Ajax 底層，同時以手動及自動的方式測試它。我們會攔截客戶端請求，測試伺服端邏輯，反之亦然，竄改伺服端回應，測試客戶端程式碼。

第 12 章，多面向測試，聚焦在期程、期程管理、以及你的安全測試如何攻擊它，並且提供一些祕技，讓你知道如何尋找、分析、最後測試出期程管理的強度。

本書印刷體裁

參照到 Unix 風格的指令稿或命令時，我們同時運用印刷體裁和一般 Unix 文件說明慣例，在內文中提供額外的資訊。參照到 Windows 導向的指令稿或命令時，我們使用印刷體裁和 Windows 使用者所熟悉的文件說明慣例。

印刷體裁

純文字

表示選單標題、選單選項、選單按鈕、及鍵盤輔助鍵（諸如 Alt 和 Ctrl）。

斜體字

表示新術語、系統呼叫、URL、主機名稱、及電子郵件位址。

定寬字

表示命令、選項、旗標、變數、屬性、鍵、函式、型別、物件、HTML 標籤、巨集、檔案內容、命令輸出、檔案名稱、副檔名、路徑名稱、及目錄。

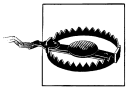
粗體定寬字

表示應該由使用者逐字輸入的命令或其他文字。

斜體定寬字

表示應該由使用者提供的替換文字。

這個圖示代表技巧、建議、或一般說明。



這個圖示代表警告或注意事項。

有時候，注意這些印刷體裁是很重要的，因為它可以區別兩個相似但不相同的概念。例如，我們經常在「解法」裡使用 URL，大多數時候，URL 都是虛構的，或者是網際網路上通用的範例 URL：`http://www.example.com/`。注意定寬字 URL 與斜體字 URL（例如，*http://ha.ckers.org/xss.html*）之間的差別，前者不是你實際應該存取的 URL（那裡實際上並沒有什麼東西），後者則是一個有用的資源，讓你實際進行參照。

範例約定

你會在範例中看到兩個不同的提示字元，我們遵循歷史悠久的 Unix 約定，使用 % 表示「非 root」的 shell（一般用戶），使用 # 表示「root」的 shell（root 或具有 root 特權的用戶）。出現在 % 提示字元後面的命令能夠（或許應該）被不具特權的用戶所執行，出現在 # 提示字元後面的命令必須為具有 root 特權的用戶所執行。範例 1 使用四個不同的命令說明這個要點。

範例 1. 使用不同提示字元的幾個命令

```
% ls -lo /var/log
% sudo ifconfig lo0 127.0.0.2 netmask 255.255.255.255
# shutdown -r now
C:\> ipconfig /renew /all
```

`ls` 命令以一般用戶的身分執行。`ifconfig` 命令以 root 的身分執行，但因為一般用戶使用 `sudo` 命令暫時提升他的特權，因此第二個命令也可以執行。第三個命令顯示 # 提示字元，假定在執行 `shutdown` 命令之前，你已經成功切換成 root 身分。

在 Windows 環境裡，我們假設你能夠在需要時啟動 CMD.EXE，打開命令提示字元，並且執行命令。範例 1 當中的 ipconfig 命令顯示出典型 Windows 命令在範例中看起來的模樣。

使用程式碼範例

這本書是為了協助你把工作做好。一般而言，你可以在你的程式和說明文件中使用本書的程式碼，除非你要重製重要的程式碼，否則無需取得我們的許可。例如，使用本書的程式碼片段撰寫程式，不需要取得我們的許可，但是，把 O'Reilly 書籍的程式範例燒成光碟片販售或散佈，就必須取得授權。引用本書的文句和範例程式碼來回答問題，不需要取得許可，但是，把本書大量的範例程式碼整合到你的產品的說明文件中，則必須取得授權。

雖然並非必要，但註明引用來源，我們會很感謝。註明來源通常包括書名、作者、出版商、以及 ISBN。例如，「*Web Security Testing Cookbook* by Paco Hope and Ben Walther. Copyright 2009 Brian Hope and Ben Walther, 978-0-596-51483-9」。

如果你覺得對本書程式碼範例的使用有別於上述情況，不用客氣，請盡管和我們連絡：*permissions@oreilly.com*。

連絡我們

在本書專屬的網頁中，我們列出勘誤表及其他額外資訊。網頁如下：

http://www.oreilly.com/catalog/9780596514839

或

http://www.oreilly.com.tw/product2_web.php?id=a244

關於本書的意見，或者想要詢問技術性問題，請寄送電子郵件到：

bookquestions@oreilly.com

或

mail@oreilly.com.tw

有關其他書籍、研討會、資源中心、及 O'Reilly Network 的資訊，可以參考我們的網站：

http://www.oreilly.com

誌謝

這本書憑藉著許多人的幫忙才有可能完成，當中有些人直接給予本書重大的協助，有些人則以不可或缺，但表面上卻不明顯的方式默默進行，我們想要在此表達對他們的由衷感謝。

Paco Hope

沒有人是一座孤島，至少對我來說是這樣。沒有來自多方人士的協助與啟發，這本書是絕對無法完成的。首先要感謝老婆大人，Rebecca，謝謝她悉心照料執行 Mac OS 以外的所有一切（像是小孩、家事、和寵物），此外，她更是一位處理不良輸入、未預期輸出、以及緩衝溢流的大師，為我們的家庭營造了溫馨和諧的氣氛。

我要感謝 Cigital 的同事與客戶，謝謝他們帶領我走進以風險為基礎的軟體安全、品質保證、及測試方法中。很多 Cigital 的同事對我的軟體安全與測試方法一直有著持續不斷的影響，下面的感謝名單係按照姓氏的字母順序反向排列：John Steven，Amit Sethi，Penny Parkinson，Jeff Payne，Scott Matsumoto，Gary McGraw，以及 Will Kruse。感謝 Alison Wade 和 SQE（Software Quality Engineering）的朋友們，謝謝

他們提供機會，讓我在他們的軟體品質研討會中講課，並且遇見一群致力提升技術的專業人士。最後，要特別提一下 Bruce Potter，謝謝他幫助我踏上寫作之路；這個傢伙真的是太棒了。

Ben Walther

Paco Hope 有眼光，有魄力，有見識，並且是這本書背後的驅動力量。有看到那幾個讀起來不像教科書的章節嗎？那些是他寫的。謝啦，Paco — 為這本書的稿酬、壓力、撰寫工作、和技術上的建議。

Cigital 的同事們，感謝你們的協助、指導、與幽默感 — 特別是辦公室裡的那幾個活寶。

最後，要讚嘆任何閱讀這本書的人，持續不斷的學習是我生命中最高的理想之一 — 你願意花時間擴展你的知識，說明你高度重視你自己的專業能力與個人原則。歡迎你提出對本書的意見與看法（要是能夠提出具體的觀點更好） — 我的 email 是 root@benwalther.net，或者，你也可以在我的部落格上留下你的寶貴意見：http://blog.benwalther.net。

給我們的審閱者

感謝技術審閱者們給予本書的所有回饋，他們的專業建議與寶貴意見確實讓我們如履薄冰地撰寫這本書，並且讓這本書變得更好。感謝 Mike Andrews、Jeremy Epstein、Matt Fisher、以及 Karen N. Johnson。

給歐萊禮團隊

最後要感謝歐萊禮的朋友們，特別是 Mike Loukides、Adam Witwer、Keith Fahlgre、和幾個才華洋溢並且協助將本書附諸實現的人。沒有 Adam 的 DocBook 以及 Keith 的 Subversion，這本書恐怕只是一堆零零落落的殘篇斷簡。