為了充分發揮 Linux 駭客系統的功能，你必須瞭解環境變數，並且熟悉如何管理它們，以獲得最佳效能、讓工作更方便，甚至隱藏蹤跡。環境變數是內建於系統和介面之中，可供整個程序使用的變數，它們控制了系統的外觀、行為和使用體驗，而且會被所有子 shell 或子程序繼承。另一方面，*shell* 變數通常使用小寫字母來表示，而且只在設定它們的那一個 shell 裡有效。

在 Linux 新手覺得困難的主題中，管理使用者環境變數可能是最難掌握的一個。本章將介紹一些最有用的環境變數和 shell 變數管理技巧，但不會過度深入探討它們之間的差異。在 Kali Linux 裡，你的環境就是你的 bash shell。每一個使用者，包括 root，都有一組預設的環境變數，其作用是決定系統的外觀、行為，和使用者體驗。你可以更改這些變數的值來讓系統運作得更有效率，並依照個人需求調整工作環境，甚至在必要時掩蓋行蹤。

將預設 shell 改為 bash

接下來，我必須深入探討一些 Linux 的艱澀知識，在開始之前，我想先跟你說聲抱歉，我會盡量簡單扼要地說明。

我們常說的終端機，嚴格說來，就是一個「shell」。它透過命令列來提供我們存取作業系統（這裡是 Linux）的管道。Linux 幾乎從誕生之初，就使用 Bourne-Again SHell（BASH）作為預設 shell 了，它不是唯一的 shell，但多數人喜歡它，並已經習慣它的特性和怪癖（quirk）了。近年來，有一個名為 *Z shell*（*zsh*）的 shell 漸受重視，它提供一些新功能，以及 bash shell 沒有的特性。Apple 和 Kali 的開發者都已經採用 zsh 了，它現在也是 Kali 的預設 shell。

本書的主要目的是以簡潔易懂的方式傳授 Linux。除了 Kali 之外，bash 是你會在幾乎大部分的其他 Linux 版本中使用的 shell，因此，我認為此時最好的做法，是將 Kali 系統的預設 shell 改成更普遍的 bash。幸運的是，Kali 的開發者讓你可以輕鬆地從預設的 zsh 切換到經典且穩定的 bash，我們來試試。

在 Kali 提示字元後面輸入 **kali-tweaks**：

```
kali> sudo kali-tweaks
```

按下 ENTER 後，你會看到一個簡單的圖形介面，如圖 7-1 所示，它可能會讓你不禁想起 Windows 98 時代（簡潔的設計搭配樸素的色彩）。

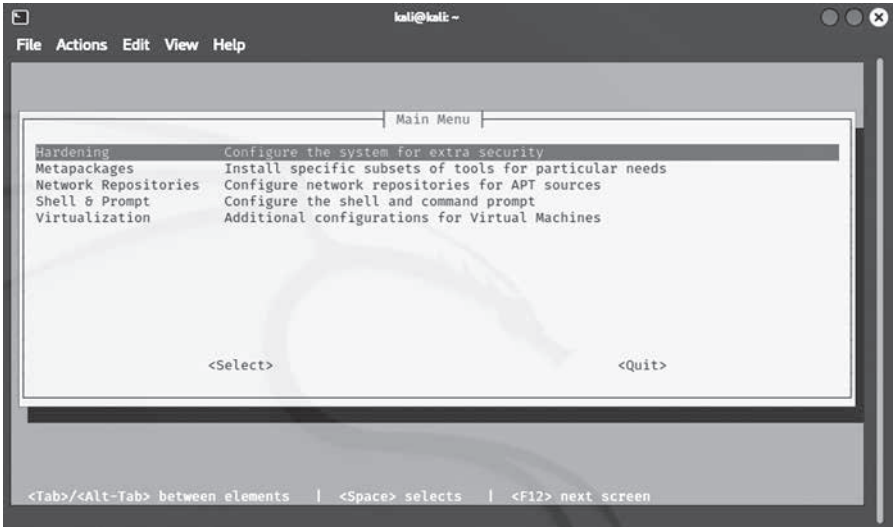

圖 7-1：kali-tweaks

使用這個選單可以強化命令列介面的安全性、安裝其他的駭客工具、設定你的套件庫，變更你的 shell 和提示字元，以及配置虛擬機器。

使用鍵盤方向鍵將光條往下移到 **Shell & Prompt**，然後按下 ENTER 來選擇它。下一個畫面提供三個選項：Configure Prompt、Set the Default Login Shell 和 Reset the Shell Config files。選擇第二個選項，按下 ENTER。

現在你會看到兩個預設的登入 shell（login shell）選項，bash 或 Z shell。將光條移到 **bash**，按空白鍵選取它，然後按下 **Apply**。關閉 shell 並登出。當你重新登入時，就會進入 bash shell。

查看與修改環境變數

你可以在終端機裡的任何目錄中輸入 env 來查看所有的預設環境變數：

```
kali> env
XDG_VTNR=7
SSHAGENT_PID=922
XDG_SESSION_ID=2
XDG_GREETER_DATA_DIR=/var/lib/lightdm/data/root
GLADE_PIXMAP_PATH=:echo
TERM=xterm-256color
SHELL=/bin/bash
--省略--
USER=kali
--省略--
PATH=/usr/local/sbin :usr/local/bin:/usr/sbin:/sbin/bin
--省略--
HOME=/kali
--省略--
```

如上所示，變數就是一對鍵值，其型態為字串。一般來說，每一對鍵值的格式都是 *KEY=value*，但如果變數有多個值，它的格式會是：*KEY=value1:value2*。和 Linux 的大多數事物一樣，如果值包含空格，你就要用引號將它框起來。

環境變數一定是大寫，例如 HOME、PATH、SHELL 等。在範例中展示的只是系統預設的環境變數。使用者也可以建立自己的變數，你將看到，你必須使用不同命令才能讓它們顯示在輸出裡。

查看所有環境變數

set 命令可以用來查看所有環境變數，包括 shell 變數、本地變數、shell 函式（例如使用者自訂變數和命令別名）。這個命令會列出系統的所有單獨的環境變數，輸出通常很長，無法在一個畫面內全部顯示。在使用 set 命令時，你可以將它 pipe 至 more 命令，以更易讀的方式逐行查看每一個變數，例如：

```
kali> set | more
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdlist:complete_fullquote:expand_aliases:extglob...
BASH_ALIASES=()
BASH_ARGC=([0] = "0")
BASH_ARGV=()
--省略--
```

現在環境變數清單會一行一行填滿一個畫面，然後暫停。當你按下 ENTER 時，終端機會往下移動一行並顯示下一個變數，因此你可以按下或按住 ENTER 來瀏覽整份清單。如第 2 章所述，當你使用 more 命令來輸出時，可以輸入 q 來離開，並回到命令提示字元。

濾出特定變數

雖然一起使用 set 與 more 比單獨使用 set 來瀏覽大量的變數名稱更容易操作和閱讀，但如果你只想要尋找特定變數，這個組合仍然不方便。你可以使用過濾命令 grep 來找出想要的變數。

我們以變數 HISTSIZE 為例。這個變數儲存的是你的命令歷史紀錄檔案最多可以儲存多少命令。命令歷史檔案存有你在當下的工作階段執行過的命令，你可以使用上下方向鍵來顯示它們。注意 HISTSIZE 沒有儲存命令本身，而是最多可以儲存幾個命令。

你可以將 set 的輸出 pipe 至 grep 以找出 HISTSIZE：

```
kali> set | grep HISTSIZE
HISTSIZE=1000
```

如你所見，這個命令找到變數 HISTSIZE 並顯示它的值。在預設情況下，它應該會被設為 1000，代表終端機會儲存你最近輸入的 1000 個命令。

在工作階段更改變數值

接著來看看如何更改變數的值。如前所述，HISTSIZE 變數儲存的是「歷史紀錄檔最多可以儲存幾個命令」。有時你不希望系統保存過去的命令，也許是因為你不想在系統上留下活動的蛛絲馬跡，在這種情況下，你可以將 HISTSIZE 設為 0，讓系統不儲存任何命令。因為這個變數只有一個值，你可以用熟悉的方式直接指定新值，如範例 7-1 所示：

```
kali> HISTSIZE=0
```

範例 7-1：更改 HISTSIZE 的值

現在當你使用上下方向鍵來取回命令時，什麼事都不會發生，因為系統不再儲存命令了。這種做法雖然可以隱藏蹤跡，卻可能帶來不便。

就像使用 Linux 的其他人一樣，駭客通常也有一些工作（*job*）、腳本或其他任務需要定期執行。舉例來說，你可能想要定期自動備份系統的檔案，或是像第 11 章那樣輪替紀錄檔。駭客也可能想要讓他們的系統每天晚上，或在他們上班或上課的時候，自動執行第 8 章的 *MySQLscanner.sh* 腳本。以上都是可以安排自動執行工作的情境。安排工作的執行時間可以讓你不需要分心執行任務，你也可以安排工作在你不使用系統的時段執行，如此一來，當你使用系統時，就有更多資源可用。

Linux 系統管理員（或駭客）也可能想要在系統開機時自動啟動某些腳本或服務。在本章中，你將學到如何使用 `cron` daemon 和 `crontab` 來讓腳本自動執行，即使系統無人看管。你也會學到如何設置啟動腳本，讓它在系統開機時自動執行，並且在你忙著執行駭客行動時，為你提供必要的服務。

安排事件或工作自動執行

`cron` daemon 與 `cron` 表（`crontab`）是在安排定期執行的工作時最實用的工具。`crond` 是一個在背景執行的 daemon。`cron` daemon 會檢查 `cron` 表來決定在指定時間要執行的命令。我們可以修改 `cron` 表來安排某項任務在特定的日期或每天的某個時間、每幾週，或每幾個月定期執行。

若要安排這些任務或工作的執行時間，你可以將它們輸入 `cron` 表，存有這個表的檔案位於 */etc/crontab*。`cron` 表有七個欄位：前五個欄位用來安排任務執行時間，第六個欄位用來指定使用者，第七個欄位儲存你想執行的命令的絕對路徑。如果要使用 `cron` 表來安排某個腳本的執行時間，你只要在第七欄填入該腳本的絕對路徑即可。

前五個時間欄位分別代表不同的時間元素：分、時、當月的幾日、月份，以及星期幾（依此順序）。每一個時間元素都必須用數字來表示，例如三月必須輸入 3（不能直接輸入「March」）。星期幾是從 0 開始計算的，0 代表星期日，最後一個數字 7 也是星期日。表 16-1 是各種時間元素。

表 16-1：crontab 使用的時間表示法

欄位	時間單位	數值範圍
1	分鐘	0–59
2	小時	0–23
3	當月的幾日	1–31
4	月份	1–12
5	星期幾	0–7

舉例來說，如果我們寫了一個腳本來掃描全世界有漏洞的開放連接埠，並希望它在每週一到週五的 2:30 AM 執行，我們可以將它排入 *crontab* 檔案。等一下會逐步說明如何將這些資訊寫入 *crontab*，在那之前，我們要先討論需要遵守的格式，如範例 16-1 所示。

```
M  H  DOM  MON  DOW  USER  COMMAND
30 2  *    *    1-5  root  /root/myscanningscript
```

範例 16-1：排程命令的格式

crontab 檔案會貼心地為你標示每一個欄位。請注意：第一欄是分鐘（30），第二欄是小時（2），第五欄是星期幾（1-5，代表週一到週五），第六欄是使用者（root），第七欄則是腳本的路徑。第三、四欄是星號（*），代表無論是當月第幾日或哪一個月份，都要在週一到週五執行這一個腳本。

在範例 16-1 中，第五欄在數字之間加上短線（-）來定義從星期幾到星期幾。若要在一週之中的幾個不連續的日子執行腳本，可以使用逗號（,）來分隔那些日子，比方將週二和週四寫成 2,4。

編輯 *crontab* 的命令是 crontab 加上 -e（edit）選項：

```
kali> crontab -e
Select an editor. To change later, run 'select-editor'.
1. /bin/nano   <----easiest
2. /usr/bin/mcedit
3. /usr/bin/vim.basic
4. /usr/bin/vim.gtk
5. /usr/bin/vim.tiny
Choose 1-5 [1]:
```

第一次執行這個命令時，系統會詢問你要使用哪一個編輯器，預設編輯器是 */bin/nano*，也就是在輸出訊息裡被標上 easiest 的那一項。如果你選擇這個選項，終端機會直接開啟 *crontab* 檔案。

另一個選項比較適合剛接觸 Linux 的新手，也就是直接使用你最習慣的文字編輯器來開啟 *crontab* 檔案，例如：

```
kali> sudo mousepad /etc/crontab
```

這個命令會用 mousepad 來開啟 *crontab* 檔案。範例 16-2 是這個檔案的部分內容。

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab, you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# which no other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user command
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron II ( cd / && run-parts
47 6 * * 7 root test -x /usr/sbin/anacron II ( cd / && run-parts
52 6 1 * * root test -x /usr/sbin/anacron II ( cd / && run-parts
#
```

範例 16-2：在文字編輯器中打開 *crontab* 檔案

現在只要新增一行設定，並儲存這個檔案，即可設定一個定期自動執行的新任務。

安排備份任務的執行時間

我們先從系統管理員的角度來討論這一項工具。身為系統管理員，你應該想要在下班後，系統閒置、資源充足的時段備份所有的檔案（系統備份往往需要消耗在工作時段特別吃緊的系統資源）。理想的備份時間可能是週末的深夜，你不會想在週六深夜 / 週日凌晨兩點登入系統（我很確定那時你一定有優先順序更高的事情要做），所以可讓系統在那個時間自動備份，即使你本人不在電腦前。

請注意，小時欄位使用 24 小時制，而不是 AM 與 PM，所以 1:00 PM 必須寫成 13:00。還要注意的是，星期幾（DOW）是從星期日（0）開始，到星期六（6）結束。

建立一項工作很簡單，只要依照指定格式，在 *crontab* 檔案中加入一行設定即可。所以，假如你要使用一個名為「backup」的使用者帳號來建立一個定期備份工作，你會寫一個備份系統的腳本，在 */bin* 目錄中將它存為 *systembackup.sh*，然後在 *crontab* 中加入下面這一行內容，安排這項備份工作在每週六深夜 / 週日凌晨 2:00 執行：

```
00 2 * * 0 backup /bin/systembackup.sh
```

請注意，萬用字元 * 代表「任何」，在當月日期、月份或星期幾欄位中使用它代表「每一」天或月。這一行命令可以按照下面的方式從左到右解讀：

1. 在整點（00 分），
2. 在第二個小時（2 點），
3. 在每個月的任何一天（*），
4. 在任何一個月份（*），
5. 在星期日（0），
6. 以 backup 使用者身分，
7. 執行位於 */bin/systembackup.sh* 的腳本。

接下來，cron daemon 就會在每個月的每個星期日 2:00 AM 自動執行這個備份腳本。

如果你只想要在每個月的 15 日與 30 日執行備份，無論那兩天是星期幾，你可以這樣修改 *crontab* 的設定：

```
00 2 15,30 * * backup /root/systembackup.sh
```

注意，我們將「當月第幾日」（DOM）欄位改為 15,30。這指示系統只在每個月的 15 日與 30 日執行該腳本，大約是每兩週一次。如果你想要指定多個日期、小時或月份，就要使用逗號來隔開它們，如同我們的寫法。

接著，假設公司要求你必須特別謹慎地執行備份，即使遇到停電或是系統當機，也絕對不能遺失任何一天資料，這種情況下，你可以加入以下這一行，在每一個平日晚上執行備份：

```
00 23 * * 1-5 backup /root/systembackup.sh
```