作為駭客，我們是天生的身體力行者。我們無時不想著摸索把玩些什麼，有時也想要創造，甚至破壞些什麼。而當中不乏有些人在栽進最愛的駭客活動前偏好先閱讀一些資訊科技理論當作行前準備。考慮到這一點，本章就是為此所設計，接下來你將學到幫助你快速上手 Kali 的基礎技巧。

本章中，我們不會對概念作太多細節的討論，而僅會涵蓋剛好足夠的範圍，讓你能自由地探索這個駭客的 Linux 作業系統。更深入的討論我們將留給之後的篇章。

專有名詞與觀念介紹

在進入給新手駭客的 *Linux* 基礎的美妙世界之前，我想先介紹並解釋一些專有名詞，這有助於稍後本章中一些觀念的討論。

二進位檔案（Binaries） 此名詞指可被執行的檔案，類似於 Windows 中的執行檔（executables）。二進位檔一般位在 */usr/bin* 或 */usr/sbin* 目錄中，例如指令工具 `ps`、`cat`、`ls`、與 `cd`（本章稍後會一一介紹），應用程式像是無線駭客工具 aircrack-ng、與入侵偵測系統 Snort 都屬於二進位檔。

大小寫敏感（Case sensitivity） 有別於 Windows，Linux 區分大小寫。舉例來說，*Desktop* 不同於 *desktop*，也不同於 *DeskTop*，它們任何一個都代表著不同的檔案或目錄名稱。這個特性常讓習慣 Windows 環境的使用者感到困擾，當你有把握某個檔案或目錄確實存在，但系統卻顯示“找不到檔案或目錄”的錯誤訊息時，你八成需要先檢查大小寫。

目錄（Directory） 等同於 Windows 中的資料夾。目錄以樹狀階層的方式存在，提供檔案的組織整理。

家目錄（Home） 每個使用者都有自己的家目錄 */home*，這也是使用者新建檔案的預設存放目錄。

Kali Kali 是 Linux 眾多發佈版本（distribution）中專門針對滲透測試所設計的版本。它已預先搭載數百個駭客最常使用的工具，替使用者省去了一一手動下載安裝各工具的時間和麻煩。書中使用的 Kali 為我在撰寫本書時的最新版：Kali 2018.2，釋出於 2018 年 4 月。

root 帳號 就像所有的作業系統，Linux 也有一個設計給被信任的人員使用的系統管理員或超級使用者帳號。該帳號能對系統做任何事，包括了重新配置系統、添加使用者或更改密碼等，在 Linux 中這個帳

終端機啟動了指令列環境，也就是 *shell* 程式，它讓你能在作業系統中執行指令並編寫腳本。Linux 有著各式各樣的 shell 環境，其中最受歡迎的是 bash shell，它也是 Kali 和許多 Linux 發佈版本的預設 shell。

如欲更改使用者密碼，請使用 `passwd` 指令。

Linux 檔案系統

有別於 Windows，Linux 不以實體磁碟（如磁碟 *C:*）作為檔案系統架構的基礎，而是採用邏輯式的檔案系統。在檔案系統架構最上層為 /，因為整個系統架構就像是棵上下顛倒的樹（見圖 1-4），/ 也被稱為檔案系統的根目錄（*root*）。請留意此處 root 指的是根目錄而不是 root 使用者，這些術語起先可能令人覺得容易混淆，但當你習慣 Linux 後將變得容易區別。

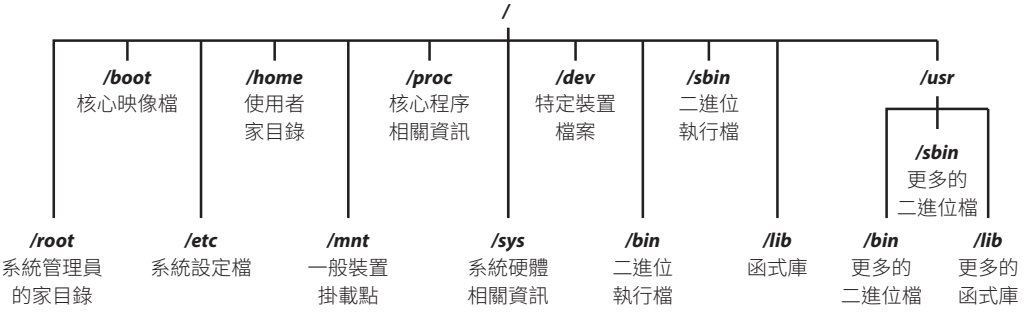

圖 1-4：Linux 檔案系統

檔案系統的根目錄（/）位於目錄樹的最上方，在它之下有幾個你必須認識的重要子目錄：

/root root 使用者的家目錄

/etc Linux 系統設定檔案一般位於此目錄，系統設定檔決定程式何時且如何啟動

/home 使用者的家目錄

/mnt 其他檔案系統接附或掛載到主檔案系統的地方

/media CD 與 USB 裝置一般於此接附或掛載到主檔案系統

/bin 應用程式的二進位檔所在處（等同於 Microsoft Windows 的執行檔）

/lib 函式庫（*libraries*）所在目錄（類似於 Windows DLLs 的共用程式）

```
kali >rmdir newdirectory
rmdir:failed to remove 'newdirectory': Directory not empty
```

值得注意的是，`rmdir` 僅能移除未含有任何內容的空目錄，否則系統將回傳如例中的錯誤訊息：Directory not empty。在移除目錄前，首先你必須移除目錄中所有內容，這是為了防止你誤刪不打算刪除的檔案。

如果想要一次性地將目錄與其內容刪除，可以使用 `rm` 指令加上 `-r` 選項來完成：

```
kali >rm -r newdirectory
```

希望你將以下這段警告牢記在心：請謹慎小心地使用 `rm -r` 指令，尤其是在你剛開始熟悉 Linux 時，一個分神或操作上的錯誤，很有可能失手刪掉寶貴的資料夾與檔案。舉例來說，如果你在家目錄執行 `rm -r`，將會直接刪除所有的目錄與檔案，相信這應該不是你所樂見的。

放膽去玩吧！

現在你已具備在 Linux 檔案系統中遊走的基本能力了，在開始學習更多課題前，你可以花點時間實際應用每個學到的技巧。熟悉終端機操作最好的方式就是試著練習每個新學到的技巧。在接下來的章節中，我們將更深入地探索屬於駭客們的遊樂場。

EXERCISES

在進入第二章之前，試著使用本章學到的技巧完成以下練習：

1. 用 `ls` 指令從根目錄（`/`）探索 Linux 的檔案結構。以 `cd` 指令移動到每個目錄後執行 `pwd` 指令來確認當前目錄所在位置。
2. 用 `whoami` 查看你所登入的使用者身分。
3. 用 `locate` 指令找到可用於密碼破解的 wordlists 檔案。
4. 先用 `cat` 指令新增一個檔案後再以 `cat` 為其添加內容。記住 `>` 用來將輸入重新導向至檔案，`>>` 則用來添加內容。
5. 建立一個名為 *hackerdirectory* 的目錄，並在其中新增一個名為 *hackedfile* 的檔案。接著將該檔案複製到 */root* 目錄，並重新命名為 *secretfile*。

在 sources.list 檔案中添加軟體套件庫

軟體套件庫（*repositories*）指的是某個 Linux 發佈版本用來存放軟體套件的伺服器。幾乎每個發佈版本都有自己的套件庫，當中的軟體也是為該發佈版本所開發或設定，因此並不是所有的軟體都能在其他發佈版環境中成功運作。雖然這些軟體庫所含有的軟體套件都大同小異，但它們並不完全相同。不同之處可能來自版本上的差異，或甚至是完全不同的軟體。

Kali 套件庫有著豐富的安全與駭客軟體，而它也將是你使用的軟體套件庫。但因為 Kali 專注在資安與駭客滲透，其套件庫可能缺少一些特定的專業軟體，甚至某些看似平凡無奇的軟體有時也不在其中。因此為了避免在 Kail 套件庫找不到某個軟體的窘境，添加一兩個備用的套件庫讓系統可做額外的搜尋是相當值得的。

你的系統用來搜尋軟體的套件庫存放在 *sources.list* 檔案中，藉由修改該檔案你可以重新定義下載軟體時系統使用的套件庫。我通常會在 *sources.list* 檔案中的 Kali 套件庫之後另加入 Ubuntu 的套件庫；如此一來，當我請求下載的軟體不存在於 Kali 套件庫時，系統會接著轉向搜尋 Ubuntu 套件庫。

你可以從 */etc/apt/sources.list* 路徑找到 *sources.list* 檔案，該檔案可用任何文字編輯器開啟。輸入以下指令開啟 *sources.list* 檔案，並將 `leafpad` 替換成你文字編輯器的名稱：

```
kali >leafpad /etc/apt/sources.list
```

指令輸入後，你應該會看到如圖 4-1 的視窗，視窗中的內容為 Kali 的預設套件庫。

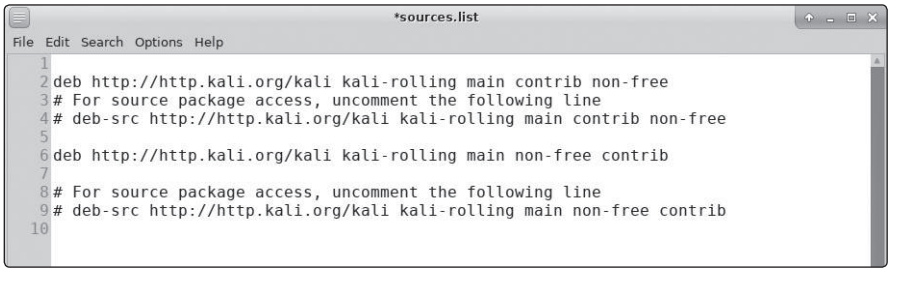

圖 4-1：在 *sources.list* 中的 Kali 預設套件庫

檢視權限

當你想知道某個檔案或目錄的使用者擁有什麼權限時，可以使用 `ls` 指令加上 `-l`（表 long）選項來顯示目錄內容中包括權限在內的詳細資訊。在清單 5-1 中，我對檔案 */usr/share/hashcat*（我最愛的密碼破解工具之一）使用 `ls -l` 指令，並一一檢視從中得到的資訊。

```
kali >ls -l /usr/share/hashcat
total 32952
❶ ❷       ❸ ❹            ❺       ❻            ❼
drwxr-xr-x  5  root  root    4096     Dec 5 10:47  charsets
-rw-r--r--  1  root  root    33685504 June 28 2018 hashcat.hcstat
-rw-r--r--  1  root  root    33685504 June 28 2018 hashcat.hctune
drwxr -xr-x 2  root  root    4096     Dec 5 10:47  masks
drwxr -xr-x 2  root  root    4096     Dec 5 10:47  OpenCL
drwxr -xr-x 3  root  root    4096     Dec 5 10:47  rules
```

清單 5-1：用 `ls -l` 指令檢視檔案的權限資訊

輸出中的每個欄位分別告訴我們以下資訊：

❶ 檔案類型
❷ 依照順序分別為擁有者、群組、其他使用者的權限
❸ 連結數（此主題超出本書範圍）
❹ 檔案擁有者
❺ 以位元組（byte）顯示的檔案大小
❻ 檔案建立或最後修改時間
❼ 檔案名稱

先讓我們將焦點放在最左方由字母和減號組成看似很難理解的字串上。此欄位提供我們判斷該項目為檔案或目錄以及其權限屬性的資訊。

第一個字元說明檔案的類型。其中 `d` 代表目錄（directory），`-`（減號）則表示檔案，這兩個也是最常見的檔案類型。

接續的字串則定義了檔案的權限。該字串以三個字元為一組，共有三組，組中的三個字元分別以 `r`（read，讀取）、`w`（write，寫入）、`x`（execute，執行）的順序與減號（`-`）交替形成不同的組合。第一組代表了擁有者的權限；第二組為群組；第三，也就是最後一組，則代表所有的其他使用者。

用遮罩設定更安全的預設權限

如先前所提，Linux 通常會自動為檔案和目錄分別配置 666 與 777 的基本權限，但你可以透過利用 umask（表 unmask）來修改檔案或目錄建立時系統配置的預設權限。umask 數值代表你想要從檔案或目錄的基本權限上移除的權限，並藉由它使檔案目錄更安全。

umask 是一個三位數數字，每個數字分別對應先前所學的三種權限數字，原始權限數字減去 umask 數字後即可決定新的權限狀態。也就是說，在一個檔案或目錄建立時，其權限會等於原始的系統預設權限減去 umask 數值後的結果，見圖 5-1。

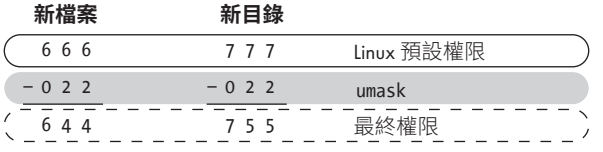

圖 5-1：值為 022 的 umask 如何對新檔案與目錄造成影響

舉例來說，當 umask 為 022 且檔案原始預設權限為 666 時，一個新檔案在建立時其權限會是 644，代表檔案擁有者有讀取和寫入的權限，而群組與其他使用者則僅有讀取權限。

在 Kali 和大部分的 Debian 系統中，其 umask 的預設值為 022，也意味著 Kali 的檔案和目錄分別有著 644 和 755 的預設權限。

umask 並不是一個對系統上所有使用者通用的值，每個使用者都可以從他們的 *.profile* 檔案為檔案或目錄設定不同的 umask 預設值。如果使用者想要查看當前 umask 值，輸入指令 umask 即可令系統回傳當前設定值。使用者也可透過編輯 */home/username/.profile* 檔案對 umask 進行修改，例如在檔案中加入 umask 007 將使得檔案擁有者以及群組成員擁有所有權限。

特殊權限

在 rwx 這三個一般用途的權限之外，Linux 還有三個較為複雜的特殊權限。這三個特殊權限分別為 SUID（也作 set user ID）、SGID（也作 set group ID）、與 sticky bit。下面三節將依序就這三個特殊權限做討論。

用 SUID 授予暫時性的 root 權限

現在你應該清楚了解到，一個使用者能否執行某檔案取決於該使用者是否擁有檔案的執行權限，若使用者僅有讀取和（或）寫入權限，他將無法執行該檔。雖然這似乎看起來理所當然，然而該規則仍存在著例外。

你可能碰過下列情形：某個檔案在執行過程中要求所有執行者擁有 root 使用者的權限，即便該使用者並非 root。舉例來說，一個允許使用者修改密碼的程式在執行過程中將需要對 */etc/shadow* 檔案進行存取，但由於該檔案儲存使用者在 Linux 系統的密碼，因此會要求進行存取的程式具有 root 權限。這時候，你可以藉由設定程式的 SUID 位元來暫時升級執行者，使其具備擁有者的執行權限。

基本上 SUID 位元讓所有使用者可以以檔案擁有者的權限執行檔案，但升級的權限僅限於該檔案的使用範圍內。

設定 SUID 位元時你需要在一般權限之前輸入 4，將檔案權限從 644 更改為 4644 來完成 SUID 的設定。

通常一般使用者並不會需要對檔案進行 SUID 的設定，但如果你想這麼做，你需要使用 chmod 指令來完成，像是 chmod 4644 *filename*（檔案名稱）。

用 SGID 授予 root 使用者的群組權限

SGID 同樣地用在將權限暫時升級，不同之處在於它給予的是擁有者所在群組的權限，而非擁有者的權限。換句話說，當 SGID 位元設定後，如果檔案擁有者的群組擁有執行該檔的權限，某個原本沒有執行權限的使用者將可以執行該檔案。

SGID 設定於目錄時則略有不同：當一個目錄設有 SGID 時，在該目錄內建立的新檔案，其擁有權將屬於目錄建立者的群組而非檔案建立者的群組。當一個目錄有多個使用者共用時，這個特性將顯得特別有用，它使得群組中的所有使用者都可以執行新建的檔案，而不僅限於單一使用者。

設定時，SGID 位元是以 2 作為代表並同樣置於一般權限之前。因此當一個權限為 644 的檔案設定 SGID 後，其新權限將會是 2644。同樣地你使用 chmod 指令來設定 SGID，如 chmod 2644 *filename*（檔案名稱）。

如清單 6-1 所示，執行 ps 指令時加上 aux 選項將列出系統上所有使用者的所有程序。注意到你無需在這些選項前加上減號（-）且它們都為小寫字母；因為 Linux 為大小寫敏感，若錯用大寫字母你將得到完全不一樣的輸出結果。

```
kali >ps aux
USER  PID   %CPU  %MEM    VSZ    RSS TTY    STAT START   TIME   COMMAND
Root    1    0.0   0.4    202540  6396 ?     Ss   Apr24    0:46  /sbin/init
Root    2    0.0   0.0         0     0 ?     S    Apr24    0:00  [kthreadd]
Root    3    0.0   0.0         0     0 ?     S    Apr24    0:26  [ksoftirqd/0]
-- 省略 --
root  39706  0.0  0.2  36096  3204 pts/0    R+ 15:05  0:00    ps aux
```

清單 6-1：利用 aux 選項查看所有使用者的程序

正如你所見，該指令列出非常多的程序，可能多到必須向上捲動視窗才能查看完整的資訊。從輸出最後的欄位我們可得知第一個程序是 init，最後一個程序則為我們剛執行的 ps aux。

許多細節的部分（PID、%CPU、TIME、COMMAND 等等）在你的系統上或許略有不同，但格式則應該相同。為了之後的討論，這邊列出指令輸出中最重要的幾個欄位：

USER 呼叫該程序的使用者

PID 程序 ID

%CPU 該程序使用多少百分比的 CPU

%MEM 該程序使用多少百分比的記憶體

COMMAND 啟動該程序的指令名稱

一般來說，當要對程序執行任何動作時，其 PID 也必須明確指出。接著就來看看我們可以如何利用這個程序識別碼。

以程序名稱篩選

當我們想對程序執行動作或做更多的了解時，我們通常不希望指令一次列出所有的程序，因為這只會讓螢幕充滿了太多不相干的資訊。多數時候我們只想查看單一特定程序的資訊，這時我們可以使用第一章介紹的 grep 指令做過濾篩選。

接著我們將使用 Metasploit 框架作示範。Metasploit 是最受廣泛使用的滲透測試框架，同時也是所有駭客的好夥伴。Kali 系統已預先安裝該框架，

藉由降低其優先權來相對地提高其他程序的優先權和可得資源，你可以如下用 renice 指令對 slowprocess（PID 6996）設定更高的 nice 值：

```
kali >renice 20 6996
```

與 nice 一樣，僅有 root 使用者能夠令 renice 接受負數值以提高程序優先權，但任何使用者都可以用 renice 提高 nice 值來降低程序優先權。

另外你也可以利用 top 工具更改程序的 nice 值。在 top 運行時輸入 r（小寫）後接著提供 PID 與 nice 值即可。清單 6-4 為我在 top 工具中輸入 r 並提供 PID 與 nice 值後的輸出結果：

```
top - 21:36:56 up 21:41,  2 users, load average: 0.60, 0.22, 0.11
Tasks: 128 total,  1 running, 127 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.5 us, 0.7 sy, 0.0 ni, 96.7 id, 1.1 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem:  511864 total,  500780 used, 11084 free,  152308 buffers
KiB Swap:  901116 total,  14444 used,  886672 free,  171376 cached
❶ PID to renice
|
PID   USER  PR  NI  VIRT   RES   SHR   S  %CPU  %MEM  TIME      COMMAND
5451  root  20  0   1577m  19m   14m   S  5.3   3.9   42:46.26  OLLYDBG.EXE
2766  root  20  0   55800  20m   5480  S  2.6   4.0   1:01.42   Xorg
5456  root  20  0   6356   4272  1780  S  1.3   0.8   13:21.69  wineserver
7     root  20  0   0      0     0     S  0.3   0.0   0:30.12   rcu_sched
5762  root  20  0   174m   20m   17m   S  0.3   4.1   0:04.74   gnome-terminal
```

清單 6-4：於執行中的 top 工具修改 nice 值

在按下 r 鍵時系統會回傳 PID to renice 的提示字元詢問程序 PID ❶。輸入完成後系統會刷新輸出結果以反映新的優先權。

殺死程序

有些時候程序可能出現異常的行為並耗用過多的系統資源，最糟的情況程序甚至會凍結。出現上述行為的程序通常被稱為殭屍程序（*zombie process*）。對你來說，這樣的現象所造成最大的問題大概是原本可為其他有用程式所利用的系統資源被浪費在殭屍程序之上。

在辨識出有問題的程序後，你可以考慮使用 kill 指令來停止該程序。用 kill 殺死一個程式有許多的方法，而每個方法都有其對應的數字。

kill 指令共有 64 個不同的 kill 訊號，每個訊號的功能也略有不同。這邊我們會將焦點放在最有用的幾個訊號之上。kill 指令的語法為 kill

-signal PID，其中訊號（signal）的選項為選擇性的。當你未提供訊號旗標時，指令將使用預設值 SIGTERM。表格 6-1 列出最常見的 kill 訊號。

表格 6-1：常用的 Kill 訊號

訊號名稱	訊號值	功能描述
SIGHUP	1	也稱為*掛斷*（*Hangup, HUP*）訊號。它停止指定的程序並以相同的 PID 重新啟動。
SIGINT	2	也稱為*中斷*（*Interrupt, INT*）訊號。雖然多數時候不會有問題，但它是個不保證成功的弱訊號。
SIGQUIT	3	也稱為核心*轉儲*（*core dump*）。它終止程序並將其資訊暫存於記憶體中，最後在當前工作目錄將該資訊存檔成名為 *core* 的檔案（其原因超出本書討論範圍）。
SIGTERM	15	這是*終止*（*Termination, TERM*）訊號。它也是 kill 指令的預設訊號。
SIGKILL	9	此為強制中斷的訊號。它透過將程序的資源送到系統上的特殊裝置 */dev/null* 以強制終止程序。

你可以使用 top 指令來辨識出使用過多資源的程序。雖然多數時候它們是無害的正常程序，但有時也可能是某個你會想殺掉的惡意程序正在消耗系統資源。

當你只想以 HUP 訊號重新啟動程序時，可以在 kill 指令後加上 -1 選項，如下：

```
kali >kill -1 6996
```

當遇到殭屍或惡意程序時，有很高的機會你會想直接傳送 kill -9 的訊號來強制停止該程序。這個指令可確保程序被確實終止。

```
kali >kill -9 6996
```

如果你不知道程序的 PID，你也可以使用 killall 指令來殺死程序。這個指令以程序名稱作為參數而非 PID。

下例示範如何終止一個虛構的程序 zombieprocess：

```
kali >killall -9 zombieprocess
```

最後，你也可以在 top 工具中終止一個程式。按下 k 鍵（小寫）後輸入程序的 PID 即可終止程序。

上方腳本提示使用者輸入第一個 IP 位址、最後一個 IP 位址、和掃描的目標埠號。在蒐集了這些資訊後，腳本接著執行 nmap 掃描並回報那些在 IP 範圍中指定通訊埠處於開啟狀態的 IP 位址。相信到這你已見識到即便是最陽春的腳本也可以是一個非常強大的工具。你在第十七章會學到更多關於腳本的編寫。

常見的 Bash 內建指令

如先前答應你的，下方表格 8-1 為你列出一些有用的 bash 內建指令。

表格 8-1：bash 內建指令

指令	功能
:	回傳 0 或 true
.	執行 shell 腳本
bg	將工作移至背景
break	跳出當前迴圈
cd	移動至其他目錄
continue	繼續當前迴圈
echo	將參數回傳為標準輸出
eval	將參數做為 shell 命令執行
exec	直接執行指令而不另建立程序
exit	退出 shell
export	將變數或函式匯出至其他程式
fg	將工作移至前端
getopts	解析 shell 腳本選項參數
jobs	列出在背景中運行的工作
pwd	顯示當前所在目錄
read	讀取標準輸入
readonly	宣告變數為唯讀
set	列出所有變數
shift	將參數位置向左移動
test	測試參數
[	用於條件測試
times	顯示使用者與系統累計程序時間
trap	設觸發陷阱捕捉訊號或事件

以上指令都能夠壓縮檔案，它們之間的差別在於使用不同演算法以及有著不同的壓縮率。接著就讓我們一一來看每個指令的功能與特性。

大致上來說，compress 指令是最快的，但壓縮後的檔案也是最大；bzip2 指令為最慢但有著最小的檔案大小；gzip 指令則介於兩者之間。身為一位成長中的駭客，這三個方法你都需要學習，因為在往後接觸工具越來越多的同時，你也會遇到各式各樣的壓縮檔案。本節也將教你如何使用幾個主要的壓縮方法。

用 gzip 壓縮檔案

讓我們先從 gzip（表 GNU zip）開始，它也是在 Linux 世界中最廣泛使用的壓縮工具。輸入以下指令壓縮你的 *HackersArise.tar* 檔案（記得先確認你是否位在打包檔的目錄中）：

```
kali >gzip HackersArise.*
```

注意到我們使用萬用字元 * 來代替副檔名，這告訴 Linux 指令執行的目標為所有以 *HackersArise* 開頭並有著任何副檔名的檔案。之後的例子我們也將使用類似的語法。當用指令查看目錄詳細內容時，我們可以看到 *HackersArise.tar* 檔案已被取代為 *HackersArise.tar.gz* 檔案，另外檔案大小也被壓縮至僅僅 3,299 bytes ！

```
kali >ls -l
-- 省略 --
-rw-r--r-- 1 root root  3299 Nov 27 2018 13:32 HackersArise.tar.gz
-- 省略 --
```

我們接著可以使用 gunzip 指令來對同一個檔案進行解壓縮。gunzip 為 *GNU unzip* 的縮寫。

```
kali >gunzip HackersArise.*
```

當解壓縮完成後，該檔案將不再以 *.tar.gz* 的副檔名存於目錄中，而是回到 *.tar* 的副檔名。另外注意到檔案大小也變回原本的 40,960 bytes。你可以試著列出目錄詳細資訊做二次確認。值得一提的是，其實 gzip 指令也可用來提取 *.zip* 檔案。