第二部份

使用者的責任

第二部份，使用者的責任，提供 UNIX 主機安全性的基本介紹。其中的章節是同時寫給使用者與管理者看的。

第三章，*使用者與通行密碼*，本章與 UNIX 上的使用者帳號有關，討論的範圍包括：通行密碼的目的，通行密碼的好壞有何區別，並介紹了 *crypt()* 密碼加密系統的工作原理。

第四章，*使用者、群組與超級使用者*，描述如何利用 UNIX 的群組控制檔案與設備的存取，也會討論 UNIX 的超級使用者與其所扮演的角色。

第五章，*UNIX 檔案系統*，討論了 UNIX 檔案系統所提供的安全措施，以及如何以檔案擁有者對檔案與目錄的存取權，對一群使用者或電腦系統中所有的人作限制。

第六章，*密碼學*，討論加密與訊息摘要在安全性中所扮演的角色，包含：PGP 郵件套件與數種常見編碼方式的討論。

3

使用者與通行密碼

本章將解釋 UNIX 的使用者帳號及密碼系統，也會討論怎樣才算是一個可靠的「通行密碼」（password，後文簡稱為密碼）。

「好的密碼安全性」（good password security），是第一道防止系統被誤用之防線的一部份【註】。當有人企圖對系統進行未經許可的存取時，通常會先試著猜出合法使用者的密碼，常見的作法有兩種：將常用的密碼建成一個資料庫，然後利用這個資料庫所儲存的密碼不斷地嘗試；再竊取系統的密碼檔，試著破解這些被加密過的密碼，藉以找出真正的密碼。

當攻擊者獲得初步的存取權限之後，他或她就可以到處窺探，試著找出其它的系統漏洞，以便進一步獲取更高的使用權限。要保證系統安全的最好方法，就是一開始便阻止未經授權的使用者登入系統。這表示你必須教育所有使用者，讓他們瞭解好的密碼安全性具備的意義，並確定他們都會遵循這些規則。

註　這道防線的另一個部份是實體的安全性，也就是必須防止攻擊者在不會受到盤查的情況下，直接經由大廳把你的伺服器搬走。第十二章對此有更詳細的討論。

有時即使有可靠的密碼仍是不夠的，尤其是在這些密碼必須跨越未受保護的網路時，更是如此。在預設的網路協定和防禦措施中，這些密碼可能會被偷窺，而讓某個未經授權的人從網路上得到這個密碼。在遇到這種情況時，就需要單次密碼（one-time password）。

3.1 使用者名稱

每個使用 UNIX 電腦的人都應該擁有一個帳號（account）。帳號以使用者名稱（username）來識別，而傳統上，每個帳號也應該擁有一個密碼，以避免未經授權的存取。使用者名稱有時也稱為帳號名稱，你必須同時知道使用者名稱和密碼才能登入 UNIX 系統。舉例來說，Rachel Cohen 在大學的電腦系統中擁有一個帳號，她的使用者名稱是 *rachel*，密碼是 Iluvfred，當她想登入大學的電腦系統時，必須輸入：

```
login: rachel
password: Iluvfred
```

使用者名稱是一個識別字（identifier）：告訴系統你是誰；密碼則是一個鑑定字（authenticator）：使用這個鑑定字向系統證明你就是你所宣稱的那個人。

標準的 UNIX 使用者名稱的長度，是 1 到 8 個字符（character）【編註】。在同一部 UNIX 電腦中，使用者名稱必須是唯一的：兩個不同的使用者不可以使用同一個使用者名稱（如果有兩個使用者使用同一個使用者名稱，事實上，他們將共用同一個帳號）。標準的 UNIX 密碼的長度也是 1 到 8 個字符，不過現在某些商業的 UNIX 系統允許較長的密碼。較長的密碼通常比較安全，因為入侵者比較難猜出這種密碼。理論上來說，不同的使用者可以使用相同的密碼，不過如果真的發生這樣的情況，就表示他們都選用了一個不可靠的密碼。

編註　character 應該翻譯成「字符」；指的是印刷或書寫時所使用的文字符號。若將這些文字符號（名稱與外觀）收集在一起，就稱為 character repertoire（字符集）。

為了表達字符集中每個字符的關係，通常會將其製成表格，並根據字符在表格中的位置，為每個字符指定一個具唯一性的號碼，又稱為 character code（字符碼）。

依據字符碼的數值順序，利用某種演算方法，以數字的形式表達這些字符，就稱為 character encoding（字符編碼）；舉例來說，如果字符集中只有 256 個字符，你可以根據字符碼，將每個字符印射到範圍 0 到 255 的整數上，也就是說每個字符代表一個位元組（byte；octet）的資料量。

在同一個電腦上，一個人可以擁有一個以上的 UNIX 帳號，此時，每個帳號都可以擁有自己的使用者名稱。使用者名稱可以是任何一串你想使用的字符（除了某些例外），而且不需要對應使用者的真正名稱。

> 注意　如果使用者名稱不是以小寫字母開頭，或是包含了特殊字符（譬如，標點符號或控制字符），在某些版本的 UNIX 上會發生問題。對許多應用程式而言（例如，某些網路郵件軟體），含有某些不常見字符的使用者名稱也會發生問題。因此，許多站台要求使用者名稱只能包括小寫字母和數字，而且必須以小寫字母開頭。

就像你的朋友會用名字來辨識你，而 UNIX 則是以使用者名稱來辨識你。當你登入 UNIX 系統時，你會告知對方自己的使用者名稱，這就像在拿起電話時你會說：「喂，我是 Sabrina」一樣【註】。當某人想寄電子郵件給你時，他會在位址前面加上你的使用者名稱，因此，對擁有很多電腦的組織而言，通常會要求所有人在每台機器上使用相同的使用者名稱，主要的原因即在減少傳遞電子郵件時所造成的困擾。

在使用者名稱的選擇上，有很大的彈性。舉例來說，John Q. Random 可以選用以下任何一個使用者名稱，而且它們基本上都是可以用的：

```
john
johnqr
johnr
jqr
jqrandom
jrandom
random
randomjq
```

註　即使你不是 Sabrina，這麼說是想讓別人認為你是 Sabrina，然而，如果和你通話的人聽過 Sabrina 的聲音，那麼他就會知道你並不是 Sabrina，因此你的聲音便無法通過鑑定。

當然，John 也可以選擇一個和他真正名字毫無關聯的使用者名稱，像是 *avocado* 或 *t42*，使用與真正名字相仿的使用者名稱只是為了方便而已。

絕大多數的組織都會要求使用者名稱的長度至少有 3 個字符；只有 1 或 2 個字符的使用者名稱也是合法的，但通常不被鼓勵使用。對大部份人而言，只有一個字符的使用者名稱非常容易造成困擾，因為你很難想出使用者 i 或 x 到底是誰。兩個字符的使用者名稱在不同站台間，也很容易造成困擾：*mg@unipress.com* 和 *mg@aol.com* 是同一個人嗎？使用沒有什麼意義的名稱（像是 *t42* 和 *xp9uu6wl*）也會造成困擾，因為這些名稱都很難記得住。

某些組織會根據使用者的名字，指定使用者名稱（有時會加上姓名的第一個字母），而有些組織則會讓使用者自己決定使用者名稱。還有些組織及線上服務會指定一串隨機的字串，當作使用者名稱，不過使用者通常並不喜歡這種作法：使用者 *xp9uu6wl* 可能會因為不斷收到原本應該寄給 *xp9uu6wi* 的信而感到困擾，除非所有人都能記牢這兩個使用者名稱，否則這種情況是無可避免的。

UNIX 也擁有一些特殊的帳號，用來執行系統管理及特殊的系統功能，一般使用者並不能使用這些特殊的帳號，這在後面會提到。

3.2 密碼

當你輸入使用者名稱之後，UNIX 通常會提示你輸入你的密碼。這一節將說明在絕大部份的系統中，UNIX 如何儲存與處理密碼，以及如何選擇一個可靠的密碼。

3.2.1 /etc/passwd 檔

UNIX 以 */etc/passwd* 檔來記錄系統中所有的使用者。*/etc/passwd* 檔中包含了：使用者名稱、使用者真實的名字、辨別用的資訊，以及每個使用者的基本帳號資訊。該檔案的每一列包含了一筆資料錄，資料錄的欄位則是以冒號（:）作分隔。

你可以用 *cat* 命令來顯示 */etc/passwd* 檔，檔案的內容通常會像這樣：

```
root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
daemon:*:1:1::/tmp:
uucp:OORoMN9FyZfNE:4:4::/var/spool/uucppublic:/usr/lib/uucp/uucico
rachel:eH5/.mj7NB3dx:181:100:Rachel Cohen:/u/rachel:/bin/ksh
arlin:f8fk3j1OIf34.:182:100:Arlin Steinberg:/u/arlin:/bin/csh
```

前三個帳號（*root*、*daemon* 以及 *uucp*）是系統帳號，而 *rachel* 和 arlin 則是一般使用者的帳號。

/etc/passwd 的每一個欄位都有非常直接的涵意，表 3-1 會以上面這個範例檔中某一列為例，說明各個欄位所代表的意義。

表 3-1：/etc/passwd 各欄位所代表的意義

欄位	內容
rachel	使用者名稱
eH5/.mj7NB3dx	使用者之「被加過的密碼」
181	使用者之使用者識別碼（UID）
100	使用者之群組識別碼（GID）
Rachel Cohen	使用者的全名（也稱為 GECOS 或 GCOS 欄位）【註 1】
/u/rachel	使用者的「主目錄」（home directory）
/bin/ksh	使用者的 shell 【註 2】

註 1　當 UNIX 剛誕生時，所執行的平台是一個小型的迷你電腦。貝爾實驗室（Bell Labs）中的許多使用者，利用他們的 UNIX 撰寫批次工作，以便藉由 RJE（Remote Job Entry；遠端工作項目）在實驗室中較大的 GECOS 電腦上執行。使用 RJE 所需的使用者識別資訊儲存在 */etc/passwd* 檔中，作為標準使用者識別資訊的一部份。GECOS 的全名是 General Electric Computer Operating System（通用電子電腦作業系統）；GE 是當時製造電腦的主要廠商之一。

註 2　若此欄位是空的，並不表示此使用者沒有 shell；它的意思是此使用者將使用 Bourne shell（*/bin/sh*）當作預設的 shell。

密碼通常是以特殊的編碼格式來表示。在第八章中將有詳細的討論。密碼本身並未儲存在這個檔案中。被加密過的密碼可能儲存在其它的遮蔽（shadow）密碼檔中，第八章也將對此有詳細的說明。至於 UID 和 GID 欄位的意義將在第四章中討論。

3.2.2 /etc/passwd 檔和網路資料庫

近年來，許多公司組織已經捨棄大型的分時（time-sharing）電腦，轉而投入大型的主從式（client/server）網路。這種網路包含了許多的伺服器，以及數以百計的工作站。使用者可以使用群組中，或是整個組織中的任何一台工作站。當這些系統啟用時，相當於所有使用者在每一台工作站上都擁有一個帳號。

但不幸地，在這些大型、分散式的系統中，你無法確定每部電腦都擁有相同的 */etc/passwd* 檔，因此目前已經有許多不同的商業系統，可以經由網路取得儲存在 */etc/passwd* 檔中的資訊。

這些系統包括：

- Sun Microsystem 的 NIS（Network Information System；網路資訊系統）
- Sun Microsystems 的 NIS+
- Open Software Foundation 的 DCE（Distributed Computing Environment；分散式計算環境）
- Next 的 NetInfo

以上這些系統都能取得儲存在每個工作站之 */etc/passwd* 檔所包含的資訊，並儲存在一或多個網路伺服器中。如果你使用其中一種系統，而且希望觀看密碼資料庫的內容，就不能只是 *cat* 這個 */etc/passwd* 檔，你必須使用此系統所特有的命令。

Sun 的 NIS 服務會補足儲存在各工作站之檔案中的資訊，因此如果你使用 NIS，而且想要獲得所有使用者帳號的列表，只要輸入下面的命令就行了：

```
cat /etc/passwd;ypcat passwd
```

Sun 的 NIS+ 服務可以依據 */etc/nsswitch.conf* 的內容，而設定來補足或代換在 */etc/passwd* 檔中的使用者資料。如果你的系統所執行的是 NIS+，就可以使用 *niscat* 命令，並指定你的 NIS+ 網域，例如：

```
niscat -o passwd.bigco
```

在執行 NetInfo 的機器上，本地的 */etc/passwd* 將被忽略，並以網路版的 *passwd* 來取代。因此，如果你用的是 NetInfo，而且希望查看使用者帳號，只要輸入：

```
nidump passwd /
```

執行 DCE 的電腦則是使用一個加密過的網路資料庫系統，當作加密過的密碼及 */etc/passwd* 的替代品。然而，為了維持相容性，某些電腦還包含了可以定期建立本地之 */etc/passwd* 檔的程式。你必須詳細閱讀使用手冊，藉以獲得你所使用之特定系統的資料。

第十九章將會對這些網路資料庫系統做更詳細的討論。

許多站台，由於系統管理者害怕系統被人侵入，因此不願意使用網路資料庫管理系統。這種恐懼可能肇因於設定太過複雜，而且有時協定本身無法抵抗攻擊。在這種環境中，管理者只會保留一份主要的使用者資訊檔案，然後週期性地複製到遠端的機器（例如，使用 *rdist*）。這種作法的缺點在於，管理者通常必須介入使用者密碼或 shell 的更改動作。一般而言，你應該詢問廠商，藉以瞭解、並精通系統的設定方式；除此之外，你還可以建立其它的安全措施，就像在第二十一章與第二十二章中所提到的方式。

注意　因為有太多方式可以取得傳統上儲存在 */etc/passwd* 的資訊，因此在這本書中，我們將使用「密碼檔」或「*/etc/passwd*」，來代表各種不同系統中所有可能的方式。在程式設計的例子中，我們將使用一個稱為「*cat-passwd*」的特殊命令，這個命令可以將密碼資料庫的內容顯示到標準輸出。在一個沒有 shadow 密碼的傳統 UNIX 系統中，*cat-passwd* 可能只是 *cat /etc/passwd*；在一個執行 NIS 的機器上，它可能是 *ypcat passwd*。你也可以自己撰寫這樣的程式，只要不斷呼叫 *getpwent()* 函式，然後印出其結果就可以了。

3.2.3 認證

當你告訴 UNIX 你是誰之後，你必須證明你的身份，這個過程稱為認證或鑑定(authentication)。有三種不同的方式，可以向電腦系統證明你的身份，你可以使用其中一或多種方式：

1. 你可以告訴電腦你所知道的某些事（例如，密碼)。
2. 你可以向電腦「展示」你所擁有的東西（例如，卡片鎖)。
3. 你可以讓電腦量測你的某個東西（例如，指紋)。

然而，沒有一種系統是萬無一失的。舉例來說，藉由竊聽你的終端線路，某個人就能夠知道你的密碼；把槍口對準你，他就能得到你的卡片鎖；如果攻擊者有一把刀，你可能連手指都沒了！一般而言，越值得信賴的認證方式，使用的方式越麻煩，而且攻擊者也必須更有企圖心，才能侵入。

3.2.4 密碼是一種共享的秘密

密碼是最簡單的認證方式：密碼是你和電腦之間所共享的秘密。當你登入時，你必須輸入密碼，藉以向電腦證明你就是你所宣稱的這個人，然後電腦會確認這個密碼，符合你所指定的帳號：如果兩者相符，電腦就允許你繼續操作。

UNIX 並不會顯示你所輸入的密碼，因此如果你用的是一部列印終端機，或是有某個人在你身後看著你輸入密碼【註】，這種方式可以給你額外的保護。

要防禦想侵入系統的外來者，UNIX 的第一道防線通常就是密碼。雖然你可以在未登入的情況下就侵入系統，或是藉由網路竊取資訊，但是許多侵入事件卻都是肇因於選擇了錯誤的密碼。

註　有時稱為 shoulder surfing（偷瞄)。

3.2.5 為什麼要使用密碼？

絕大多數桌上型的個人電腦都沒有使用密碼（不過有許多協力廠商的程式，可以提供各種不同程度的保護，而且 Windows 和 Macintosh 的網路檔案系統也都使用了密碼）。不管是對機器的主要使用者，或是任何碰巧走到這個區域的外人而言，個人電腦沒有密碼的事實都讓電腦變得容易使用。個人電腦的使用者依靠實體的安全性 —— 門、牆和鎖 —— 來保護儲存在磁碟中的資料，避免被他人破壞。

同樣地，當初開發 UNIX 作業系統的許多研究小組也沒有為個別的使用者加上密碼 —— 原因通常相同，他們利用桌上的鎖和辦公室的門來保護電腦。在這種環境中，信任、尊重以及社會慣例，是遏阻資訊被竊取和破壞的最主要力量。

但當電腦接上數據機，幾乎使得世界上任何一個有電話的地方，都可以存取電腦中的資料，或是當電腦接上網路，讓小組以外的人也能使用之後，為帳號加上密碼就只像是為房子的門加上鎖一樣：沒有它們，入侵者可以直接闖入、而不會受到任何阻礙，而且可以盡情破壞。事實上，在今日這個電子世界中，有許多人在嘗試每部電腦中他們所能找到的「前門」，如果這個門沒鎖，有時就會招來災禍、並造成破壞。

對於多人共同的電腦，或是將電腦連上網路，而在這個網路上的各個電腦間有信任關係（本章稍後將有詳細的說明）時，密碼尤其重要。在這種情況下，一個容易被侵入的帳號，將會危害整個組織或網路的安全。

3.2.6 傳統的 UNIX 密碼

目前，絕大多數的 UNIX 系統都使用密碼來認證使用者：使用者知道密碼，當他輸入密碼之後就可以登入系統。

自早期的 UNIX 以來，密碼就已經是 UNIX 的一部份了，它的優點在於:執行時並不需要任何特殊的裝置（像是讀卡機或是指紋掃描器）。

傳統密碼的缺點在於它們很容易失敗，尤其是當你經由網路登入電腦時。近年來，在一個使用網路的環境中，傳統密碼已經不能提供可靠的安全性：一個有經驗的入侵者有太多機會可以竊取密碼【註】，以便伺機而動。現今，拜網路上各種精巧的工具所賜，即使不是非常熟練的破壞者也有能力發動攻擊。想要經由網路（像是 Internet），在遠端使用一部 UNIX 電腦，唯一可能保證安全的作法就是使用單次密碼，或是將資料加密（請參考本章稍後的「單次密碼」、第八章「保護你的帳號」、以及第六章「密碼學」）。

不幸地，我們生存在一個不完美的世界，而且絕大部份的 UNIX 仍須依賴可重複使用的密碼來辨識使用者，因此，對想要破壞連上網路之 UNIX 系統的人而言，密碼仍然是最被廣為採用的方法之一。

3.3 輸入你的密碼

你必須輸入你的密碼，藉以向電腦證明你就是你。在正統的說法中，電腦使用密碼來認證你的身份（「認證」（authenticate）與「身份」（identity），在安全專家眼中具有相當的重要性，但在這裡只代表一般的意義）。

當你登入時，必須在 login 提示符號後面輸入你的使用者名稱，藉以告訴電腦你是誰。接著必須輸入密碼（以回應 password 提示符號），證明你就是你所宣稱的那個人。舉例來說：

```
login: sarah
password: tuna4fis
```

就如我們曾說過的，UNIX 並不會把你所輸入的密碼顯示在螢幕上。

如果你所輸入的使用者名稱和密碼，符合檔案中的某一筆紀錄，UNIX 就會讓你登入，並讓你完全得到你所擁有檔案、命令以及裝置的使用權限。如果使用者名稱或密碼其中之一不符，UNIX 就不會讓你登入。

註　　對絕大多數獨立、且有實際連接到終端機的系統而言，密碼仍然是最有效的方法。

在某些版本的 UNIX 中，如果某個人嘗試登入你的帳號，並且連續輸入好幾次錯誤的密碼，那麼你的帳號將會被鎖住。只有系統管理者才能將被鎖住的帳號打開。將帳號鎖住有兩個功用：

1. 它可以保護系統，避免某個人不斷地嘗試猜出某個密碼；因為在他能夠猜出一個密碼之前，這個帳號已經被關閉了。
2. 它可以告訴你，某個人正在試著侵入你的帳號。

如果你發現你的帳號被鎖住了，應該立刻聯絡系統管理者，並且立刻更新你的密碼。絕不可以將密碼再換回帳號被鎖住之前所用的密碼。

> 注意　自動鎖住帳號的特性可以避免未經授權的使用，但也可能被利用而成為拒絕服務的攻擊，或是被入侵者用來將某些使用者鎖在系統外，讓他們無法發覺入侵者的行動。一個愛開玩笑的人可能用它來騷擾其它職員或學生，而且你也可能在喝下早上第一杯咖啡之前，因為試著登入太多次而將自己鎖住。根據我們的經驗，採用自動無限期的鎖住機制並無太大作用，比較好的方法是對登入採用遞增的延遲機制。當一定次數的登入失敗之後，就逐漸增加每個提示訊息之間的延遲時間。在網路環境中實作這樣的延遲，就必須記錄登入失敗的次數，因此入侵者就無法利用先切斷和目標機器之間的連接，然後再連接一次的方法，來避開這種延遲。

在 AIX 第四版中，只要編輯 */etc/security/login.cfg* 檔中的 *logindelay* 變數，就可以對連續登入之提示訊息使用遞增的延遲。而編輯同一檔案中的 *logindisable* 和 *loginreenable* 屬性時，AIX 就可以自動將終端機鎖住，並在某個固定時間後再自動打開。

Linux 作業系統讓使用者有 10 次登入的機會，並在每個嘗試之後增加延遲的時間。這種作法基本上和鎖住帳號的目的是一樣的（避免某個人在很短的時間內，嘗試許多個不同的密碼），但是它還可以限制入侵者對系統進行癱瘓服務（denial of service）攻擊。

3.4 改變你的密碼

你可以用 *passwd* 命令來改變密碼。*passwd* 會先要求你輸入原本的密碼，然後再要求你輸入新的密碼。先要求你輸入原本的密碼，是為了防止某個人直接走到你所登入的終端機前，然後在未告知你的情況下更改你的密碼。

當你改變密碼時，UNIX 會要求你將密碼輸入兩次：

```
% passwd
Changing password for sarah.
Old password:tuna4fis
New password: nosmis32
Retype new password: nosmis32
%
```

如果這兩次所輸入的密碼並不相同，那麼你的密碼將不會被更改，這是一個安全機制：如果你不小心把新密碼打錯了，而且 UNIX 只要求你輸入一次密碼，那麼你的密碼將變成一個連你自己也無法知道的字串。

> **注意** 對使用 Sun Microsystems 之 NIS 或 NIS+ 的機器而言，你必須使用 *yppasswd* 或 *nispasswd* 命令來更改密碼。除了名稱不同之外，這些命令的運作方式和 *passwd* 完全相同。然而，當這些命令執行時，它們會更改儲存在 NIS 或 NIS+ 網路資料庫中的密碼。當執行完畢之後，你可以立刻在網路中其它用戶端上使用新的密碼。若使用 NIS，你的密碼將會在下一次定期更新時被傳遞出去。

雖然你所輸入的密碼並不會顯示在螢幕上，但是 BACKSPACE 或 DELETE 鍵（或任何執行「刪除」（erase）功能的按鍵），仍然可以刪除上一次所輸入的字符。因此，如果你在輸入密碼時打錯字，仍然可以更正它。

當你更改密碼之後，原來的密碼就會失效。絕對不要忘記新的密碼！如果你忘記新的密碼，你必須要求系統管理者將密碼改成你所知道的字串，然後用這個字串登入後，再把它換成新的密碼。

如果系統管理者為你設定了一個新的密碼，你必須立刻把它換成只有你知道的密碼！否則，如果你的系統管理者習慣為健忘的使用者設定相同的密碼，那麼你的帳號就有可能被某個人侵入，請參考後面的說明方塊。

注意 如果你收到一封來自系統管理者的郵件，告訴你因為系統發生了一些問題，你應該立刻將密碼換成 tunafish（或任何其它的文字），千萬不要理會這個訊息，並且立刻通知系統管理者。這種郵件通常來自電腦駭客，目的是為了欺騙初學者，他們希望初學者能夠同意這項要求，並且更改自己的密碼—通常這會造成毀滅性的結果。

3.5 確認你的新密碼

當你更改密碼之後，必須先試著以新的密碼登入你的帳號，以確定你所輸入的新密碼是正確的。理論上來說，在做這個測試時不應該登出系統，這樣若是你並沒有正確地更改密碼，還可以有補救的機會。如果你以 *root* 的身份登入，而且剛才改變的是 *root* 的密碼，這點尤其重要。

強制更換密碼

據我們所知，在一所知名的大學中，時常發生學生更換密碼後，就無法登入帳號的問題。發生這種情形，絕大部份是因為學生們試著在密碼中加入控制字符【註】；有些時候則是因為學生把密碼拼錯了，以致無法再輸入相同的密碼；還有一些人是因為採用了太複雜的密碼，複雜到連他們自己都記不住。

註 在密碼中不應該包含控制字符（包括 ^@、^G、^H、^J、^M、^Q、^S 和 ^[），因為系統會解譯這些控制字符。如果使用者利用 *xdm* 登入，那麼就應該避免使用所有的控制字符，因為 *xdm* 通常會將這些字符濾掉。你也應該知道，控制字符可能會和終端機程式、終端機螢幕或任何你所使用的中間系統產生交互作用。最後要提醒你的是，你應該避免在密碼中使用 # 和 @ 字符，因為某些 UNIX 系統會將這些字符當作 erase 和 kill 字符。

一旦輸入密碼之後，就無法再將它解碼並復原了，唯一的方法就是要某個知道密碼的人，把它改成另一個已知的值。因此，學生必須到計算機中心，有一位工作人員會將他們的密碼改成 *ChangeMe*，然後要他們立刻到終端機室將密碼換掉。

這個學期末，也就是 Internet worm 事件剛結束時，有一位工作人員決定執行一個密碼破解程式（請參閱第八章），看看哪些學生的密碼容易被破解。這位工作人員很驚訝地發現，有好幾十位學生的帳號密碼仍然是 *ChangeMe*，除此之外，還有至少一位工作人員的密碼也是 *ChangeMe*！因此他們改變政策，要求這些健忘的學生必須立刻更改他們的密碼。

在 SVR4 中，超級使用者在執行 *passwd* 命令時，可以使用 *-f* 選項（例如 *passwd -f nomemory*），這樣就可以強迫使用者在下一次登入時就更換他的密碼。對系統管理者而言，這是一個非常有用的功能（這也是 AIX 的預設行為，而 OSF/1 則使用 *chfn* 命令，來達到相同的目的）。

測試新密碼的方法之一，是使用 *su* 命令。一般而言，*su* 命令是用來切換帳號的，但是因為這個命令會要求輸入另一個帳號的密碼，因此你可以利用這個命令來測試新的密碼。

```
% su nosmis
password: mypassword
%
```

（當然，你要輸入的並不是 *nosmis* 和 ~~mypassword~~，你應該使用自己的帳號和密碼）

如果你的機器在網路上，你就可以利用 *telnet* 或 *rlogin* 程式，經由網路再登入一次：

```
% telnet localhost
Trying 127.0.0.1...
Connected to localhost
Escape character is '^]'


artemis login: dawn
password: techtalk
Last login: Sun Feb 3 11:48:45 on ttyb
%
```

你可以把 `localhost` 換成你的機器的名字。

如果你嘗試了前面所提到的方法，但卻發現密碼和你所想的並不相同，就表示你遇到麻煩了。要將密碼換成你知道的值，你必須知道目前的密碼，然而你卻不知道這個密碼！你需要超級使用者的幫助才能解決這個問題。（這就是為什麼你不應該登出—如果現在是星期六凌晨兩點，你可能必須到星期一早上才能找到超級使用者，但你卻仍有工作要做）。

超級使用者（就是 *root* 使用者）也無法將任何使用者的密碼解開，然而當你不知道所設定的密碼時，他可以將你的密碼設成某個其它的值。如果你是以超級使用者的身份登入時，就可以設定任何使用者的密碼（包括你自己），而且不必輸入原本的密碼，只要在執行 *passwd* 命令時，以某個使用者名稱當作參數就行了：

```
# passwd cindy
New password: NewR-pas
Retype new password: NewR-pas
#
```

3.6 密碼的注意事項

雖然密碼是電腦安全最重要的元素，但使用者常常只是隨便選用一個密碼。

如果你是一個使用者，要知道：選了一個不可靠的密碼—或將密碼展示給不能信賴的人—這樣就有可能破壞整個電腦的安全性。如果你是一個系統管理者，請確定所有使用者都瞭解本節所提出的議題。

3.6.1 不可靠的密碼：沒有上鎖的門

所謂不可靠的密碼，就是容易被猜出來的密碼。

在電影 *Real Genius* 中，有個名叫 Laszlo Hollyfeld 的人利用電話線，藉由猜密碼的方式，侵入了軍方最高機密的電腦。Laszlo 的方法是先輸入密碼 *AAAAAA*，然後再輸入 *AAAAAB*、*AAAAAC*，諸如此類，直到找到符合的密碼為止。

現實世界的入侵者就聰明多了，他們不會一個一個地輸入密碼，而是利用電腦打電話（或是啟開網路連接），然後測試密碼，並在連線被切斷後繼續自動嘗試。入侵者並不會由 *AAAAAA* 開始嘗試，而是利用常見密碼的列表（像是 *wizard* 或 *demo* 這些字），即使是最簡單的家用電腦，加上一個不錯的密碼猜測程式，也可以在一天之內嘗試好幾千個密碼。某些入侵者的常見密碼列表有好幾十萬個密碼【註】，因此，任何其它人會使用的密碼，對你而言或許就是一個不可靠的密碼。

哪些密碼是常用、但不可靠的密碼呢？例如：你的名字、朋友的名字、父母的名字，把這些名字倒過來寫，或是在後面加上一個數字。密碼太短也不好，因為這樣的組合很少：因此，這種密碼也很容易被猜出來。來自於電腦遊戲的「魔術字」（magic word）尤其糟糕（例如 *xyzzy*），看起來雖然很隱密而且不容易被猜出來，但事實上大家都知道。其它不好的選擇包括：電話號碼、由你喜歡的書或電影選出來的字、本地建築的名字、常見的飲料名稱、或是有名的電腦科學家（本章稍後的說明方塊將會告訴你還有哪些錯誤的選擇）。把這些字反過來寫或變成大寫也不是個好主意，把字母 l（小寫的 L）換成 1（數字的一）、把 E 換成 3、在前面或後面加上一個數字、或是任何簡單的修改，都沒有什麼用處。另一種語言的單字也不是個好選擇，因為在 Internet 或 BBS 就可以下載好幾十種語言的字典。

許多 UNIX 的版本都會設法避免使用者選擇一個不可靠的密碼。例如，在某個 UNIX 版本中，如果你試著選用一個少於六個字母的密碼，而且這六個字母的大小寫都一樣，那麼 *passwd* 程式將會要求你說：*Please use a longer password*。然而，在嘗試三次之後，這個程式就會讓使用者選擇較短的密碼。比較好的版本還可以讓系統管理者指定密碼的最短長度、要求密碼必須包含非英文字母，或是其它的限制。然而，某些系統管理者會因為使用者的抱怨而解除這些限制，但這並不是個好主意，因為如果使用者的帳號被侵入了，他們會抱怨的更大聲。

註　如果你在一台家用電腦上執行一個程式，想由 *AAAAAA* 到 *ZZZZZ* 嘗試所有六個字母組合的密碼，那將會有 308,915,766 個不同的密碼，假設每秒可以嘗試一個密碼，那麼必須要將近十年的時間才能試完所有的密碼。

3.6.2 Joe 帳號

令人驚訝地，專家相信在所有沒有對密碼內容做控制的電腦中，有相當大比例的電腦，會包含至少一個使用者名稱和密碼完全相同的帳號，像這樣的帳號通常稱為 Joe。Joe 帳號很容易被入侵者找到，而且不費吹灰之力就能侵入。只要檢查系統中是否有 Joe 帳號，入侵者就幾乎可以侵入任何的系統。這也就是為什麼公開所有合法使用者名稱，是一件非常危險的事。

3.6.3 可靠的密碼：上鎖的門

可靠的密碼就是很難被猜出的密碼。想猜出最可靠的密碼非常困難，因為它們：

- 同時包含大寫及小寫字母
- 除了英文字母之外，還有數字以及（或）標點符號
- 可能包括某些控制字符以及（或）空白
- 很容易記，因此不需要把它們寫下來
- 長度是七個或八個字母

可以很快的輸入，因此任何人都無法在你身後看到你輸入的密碼是什麼。

要選擇一個可靠的密碼很容易，以下是一些建議：

- 選擇兩個簡短的單字，然後用一個特殊字符或數字把它們接在一起，例如 *robot4my* 或 *eye-con*。
- 使用對你有意義的首字母縮寫，像是 *Notfsw*（None Of This Fancy Stuff Works）、*auPEGC*（All UNIX Programmers eat green cheese）、或 *Ttl*Hiww*（Twinkle, twinkle, little star. How I Wonder what …）。

當然，*robot4my*、*eye-con*、*Notfsw*、*Ttl*Hiww* 以及 *auPEGC* 都已不是可靠的密碼了，因為它們都已經在這本書出現過。

不可靠的密碼

當你要設定密碼時，請避開以下的選擇：

- 你的名字、配偶的名字或是朋友的名字
- 你的寵物的名字、小孩的名字
- 最親近之朋友或同事的名字
- 最喜歡之科幻小說主角的名字
- 老闆的名字
- 任何人的名字
- 你目前所使用之作業系統的名稱
- 你在 passwd 檔記錄項之 GECOS 欄位的資訊
- 你的電腦主機名稱
- 你的電話號碼或牌照號碼
- 社會安全號碼（身份證號碼）的任何部份
- 任何人的生日
- 其它關於你而且很容易取得的資訊（例如：地址、過去就讀的學校）
- 像是 *wizard*、*guru* 或 *gandalf* 這類的字
- 在電腦中任何使用者名稱的任何形式（大寫、重複）
- 在英文或任何語言字的字典中的單字

- 地方的名稱或任何正確的名詞
- 完全由相同字母所組成的密碼
- 在鍵盤中字母的簡單樣式，例如 *qwerty*
- 將上述的密碼反過來
- 將上述的密碼後面加一個數字

密碼的數量

如果把一些不該用於密碼的控制字符除外，你仍然可能在標準的 UNIX 中建立超過 43,000,000,000,000,000 個完全不同的密碼。

結合十種不同語言的字典，再加上把這些字反向、大寫、在後面加上一個數字、以及其它稍微的修改，可能的密碼會少於 5,000,000 個字。再加上幾千個名字以及來自於流行文化的單字，也沒有多出很多。

因此，我們發現如果使用者選擇了一個不可靠的密碼，將非常容易被入侵者破解—他們可以把搜尋空間減少到所有可能密碼的 0.0000000012%。

一項研究指出，在一個完全沒有限制的環境中【註】，使用者在密碼中加入控制字符的機率只有 1.4%，加上標點符號和空白的機率小於 6%。密碼中也可以使用 !@#$%^&*()_-+=[]|\;:'"?/,.<>`~'，不過某些系統會把 \、# 和 @ 等符號視為 escape（literal）、erase 和 kill。（「強制更換密碼」的說明方塊中，列出了所有不應該包括在密碼中的控制字符。）

如果下次再有某位使用者，因為你所設定的密碼選擇規則而抱怨：「我根本想不出任何不會被這個程式拒絕的密碼」，你可以把這一頁拿給他看。

註　請參考附錄 D 的 Observing Reusable Password Choices。

3.6.4 在多部機器上的密碼

如果你在好幾台電腦上都有帳號，你可能希望在每台機器上使用相同的密碼，這樣就不必花時間記住所有的密碼了。然而，如果你在許多機器上使用相同的密碼，只要其中一台機器被侵入了，你所有的帳號都將會被侵入。對於一個在許多機器上都擁有帳號的人來說，常見的作法是，先設定一個基本的密碼，然後將這個密碼稍微修改一下，便當作不同機器上的帳號密碼。舉例來說，假設基本密碼是 *kxyzzy*，那麼你可以在後面加上機器名稱的第一個字母當作密碼；如果你所使用的電腦名稱是 *athena*，密碼就是 *kxyzzya*；假設電腦的名稱是 *ems*，密碼就是 *kxyzzye*。（當然，絕不要完全使用這種方式來修改你的主要密碼。）

3.6.5 把密碼寫下來

這是關於一個高中生如何侵入學校的電腦，並竄改成績的老故事。他先在學校的辦公室？走來走去，看看每位職員的終端機，並抄下他們電話號碼、使用者名稱以及記錄在 Post-It 短箋上的密碼。

很不幸地，這個故事是真的 —— 已經發生好幾百次了。

使用者經常被告誡「絕對不要把密碼寫下來」，理由其實非常簡單：如果你把密碼寫下來，別人就能找到它，並用它來侵入你的電腦。把密碼記在腦子？比把它寫在紙上安全得多，因為別人幾乎沒有機會可以知道你的密碼。換句話說，一個必須寫下來才能記得住的密碼，就表示它並不是一個很容易被猜到的密碼，如果你把寫著密碼的東西放在你的皮夾內，那麼想要用這個密碼侵入你的帳號的人，必須先偷到你的皮夾，然而這種機會卻非常小【註】。

註　當然，除非你是一位非常重要的人，而且你的皮夾是在一次精心策劃的計謀中被偷或是被搶。在 Cyberpunks 這本書中，作者 John Markoff 和 Katie Hafner 描敘了一位侵入軍方電腦的女人 —— Susan Thunder，她的方法是先在酒吧勾搭上辦公室？的人，然後和他們回家，當天晚上，當那些職員睡著的時候，Thunder 就起床，打開那些男人的皮夾，然後抄下他們的電話號碼、使用者名稱以及密碼。

如果你必須把密碼寫下來，那麼至少必須採取以下的預防措施：

- 當你把密碼寫下來時，不要註明它是一個密碼。
- 不要在記錄密碼的同一張紙上，寫下帳號名稱、網路名稱或是該電腦的電話號碼。
- 不要把寫著密碼的紙條貼在終端機、鍵盤或電腦的任何一個地方。
- 不可以寫下真正的密碼，你必須混合其它的字元，或是以某種你可以瞭解的文句方式，來掩飾真正的密碼。舉例來說，如果你的密碼是 Iluvfred，你可以寫 fredIluv、vfredxylu 或是 Last week, I lost Uncle Vemon's `fried rice & eggplant delight' recipe - remember to call him after 3 p.m.，以預防竊取皮夾的小偷【註】。

以下是你必須避免的事：

- 除非密碼已經經過編碼，否則不要將密碼記錄在線上（在檔案中、在資料庫中或是在一封電子郵件中）。
- 同樣地，絕不可以透過電子郵件將密碼傳遞給另一個使用者。在《The Cuckoo's Egg》中，Cliff Stoll 解釋一個入侵者，如何藉著在文字檔和電子郵件中搜尋 password 字串，成功地侵入了一個又一個的系統。藉由這個簡單的技巧，這名入侵者知道在許多不同的電腦中，很多不同帳號的密碼。
- 不要使用你的登入密碼當作其它應用程式的密碼。例如，不要用登入密碼當作 MUD（一種線上遊戲）或 WWW 帳號的密碼。這些應用程式的密碼是由其他人所控制的，而且有可能被不該看到的人所看到。
- 不要在由不同組織所維護之不同電腦中，使用相同的密碼。如果你這麼做，而且有一個攻擊者知道某個帳號的密碼，那麼你所有的帳號都將會被侵入。

最後一件應該避免的事，其實是很難遵守的。

註　最後一種方法還可以有其它的意圖，3 p.m. 表示從第三個字開始，取每個單字的第一個字母。因此你可以使用任何只有你能瞭解的方法，把密碼的真正含義隱藏起來。

3.7 單次密碼

想要將肇因於不安全密碼所帶來的危險性降到最低，最有效的辦法就是完全不使用傳統的密碼。你可以在站台上安裝軟體以及（或是）硬體，以便提供單次密碼（one-time password）。單次密碼正如其名，就是只用一次的密碼。

身為一個使用者，你會得到一份密碼的列表，每當你使用一個密碼，就把這個密碼劃掉，當你下一次再登入時，必須使用列表中的下一次密碼。你也可能得到一張電子卡片，卡片上會顯示一個數字，這個數字每分鐘會改變一次。你也可能會有一個小型計算機，當電腦要求你登入時，它會顯示一個數字，然後你將這個數字輸入你的小型計算機，再輸入你的個人識別號碼（Personal Identification Number；PIN），最後將這個小型計算機所顯示的數字輸入電腦，就可以登入了。

和傳統的系統相比，這些單次密碼系統都對系統的安全性提供了驚人的改善。但不幸地，因為它們都必須安裝特殊的軟體，或是購買額外的硬體，因此目前在 UNIX 的市場上並未廣為採用。

第八章對單次密碼有更詳細的解釋，同時也舉例說明目前市面上可供使用的一些單次密碼系統。

在不同的地方使用相同的密碼

Crack 程式（請參考第八章）的作者 Alec Muffett，曾告訴我們一個有趣的故事，內容是關於「在不同的地方使用相同的密碼」，我們現在轉述給你聽。

Alec 在學生時期有一個朋友（就叫他 Bob 吧），Bob 在一家電腦公司工作，當休假時，他會回到學校使用 Alec 的電腦玩 AberMUD（一種網路遊戲）。Bob 在公司的責任之一就是系統管理，這家公司的主要業務是與系統保全有關，因此所有的密碼就是隨機的字串，而且沒有任何可察覺的樣式或順序。

有一天，Alec 把 AberMUD 的密碼當成字典，放到他所發展的 *Crack* 程式中（因為這些密碼都以純文字的方式儲存在他的電腦中），然後用這個檔案測試系統使用者密碼的檔案，而且找到了一些學生帳號的密碼。在要求這些學生更換密碼之後，他就忘了這件事。

稍後不久，Alec 公開了 *Crack* 程式的修訂版，並將相關的檔案放到 Usenet。這些檔案流傳的速度非常快，最後，當經過了幾千哩的旅程之後，它們到了 Bob 的公司。Bob 是一個對系統管理非常有興趣的人，因此決定下載這些檔案，並用它們來檢查公司內的密碼。你可以想像 Bob 的驚訝和恐懼，流傳甚廣的 *Crack* 竟然能如此迅速地找出他隨意選擇、屬於最高機密的 *root* 密碼。

這個故事的寓義在於，你必須教導你的使用者，絕不可以把帳號的密碼用在其它的應用程式，或是同一個領域之外的系統中，因為他們永遠不知道這個密碼何時會再回來拜訪他！（任何類似 AberMUD 的程式都應該修改，因為在儲存密碼之前，應該先以單向雜湊函數編碼）。

3.8 摘要

這一章我們已經討論過 UNIX 如何在登入時識別使用者，以及如何認證使用者的身份。我們也說明了如何表示及使用密碼的細節。在稍後幾章的內容中，我們將介紹更進一步的技術資訊，讓你瞭解如何保護你的密碼檔以及密碼。但能夠保護你的系統之最基本而且最重要的忠告，可以摘錄如下：

- 如果可能的話，使用單次密碼

否則的話：

- 確定每個帳號都有密碼
- 確定每個使用者都選擇了一個不易被破解的密碼
- 不要將你的密碼告訴任何人

記住：即使全世界最厲害的電腦入侵者連接到你的機器上，當他面對 `login:` 提示訊息時，他也只能不斷地猜測使用者名稱和密碼，希望有一對組合是正確的。除非這個電腦犯是以你的機器作為復仇的目標，或是因為在你的系統中有非常特殊的資訊，否則入侵者通常會放棄，並試著侵入另一台機器。

想要建立一個安全的電腦系統，最重要的事情之一，就是確定使用者選擇了一個可靠的密碼。