
Enterprise Linux 實戰講座

RHEL 4 上的 Samba 伺服器 (一)

1991 年 Andrew Tridgwell 透過資料封包的反向解析了解微軟 SMB 協定的運作方式，而寫出 Samba 軟體，從此開啟 Linux 作業系統與微軟的作業系統互通的大門，只要在 Linux 上啟用 Samba 服務，那麼 Linux 就宛如微軟的主機，可利用 SMB 協定與 MS Windows 的主機互相連線，並達到資料分享、帳號認證、NetBIOS 名稱解析等功能。

前言

Unix 的世界利用 NFS 讓 Unix-Like 主機間彼此間可以共享資源，而同樣地，微軟為了讓 DOS/Windows 主機間可以資源共享，於 1980 年代發展出有別於 NFS 的 SMB (Server Message Block) 通訊協定，使得網路芳鄰主機間的檔案系統、印表機功能得以資源分享。

由於 Sun 公司於 1985 年將其附於 Sun 作業系統的網路檔案系統的規格完全公開，所以在許多 Unix-Like 的平台上皆可見到 NFS 的蹤影，在那個年代，Unix Like 要分享另一台 Unix 主機的資源可利用 NFS。但是若要分享 Windows 主機的資源，因為微軟未將 SMB 協定公諸於世，所以可是「無解」。

直至 1991 年，Andrew Tridgwell (現任職於 IBM Research) 透過資料封包的反向解析了解 SMB 協定的運作方式，而寫出 Samba 這個自由軟體，只要在 Linux 上啟用 Samba 服務，那麼 Linux 就宛如微軟的主機，可利用 SMB 協定與 MS Windows 的主機互相連線，並達到資料分享、帳號認證、NetBIOS 名稱解析等功能。

Samba 可以取代 NT 『網路上的芳鄰』檔案及印表機分享功能，Samba 也可以完全取代 NT PDC (Primary Domain Controller) 成為 NT 網域主控者管理 NT 網域；若是在同一台 Server 架設 Samba 及 Apache，則在辦公室或校園環境內，使用者可用自己的帳號及密碼從 Windows 登入網域，再由 『網路上的芳鄰』進入使用者個人帳號下放置網頁的目錄，進行編輯個人網頁 (傳統的方式是先個人電腦上編輯網頁，再使用 ftp 上傳)。

Samba 與 Microsoft Network 的關係

在設定 Samba 服務之前，必須先瞭解 Microsoft Network 的基本概念。因為 Samba 正是 Linux 和 Windows Network 的橋樑，它的設計是讓 Unix-Like OS 加入到 Windows Network，而不是讓 Windows Network 加入 Unix-Like OS 中，千萬不要搞混了！其實讀者可以這樣想，當 Linux 啟動 Samba 服務後，這台 Linux 主機就像是一台 Microsoft 的主機。

首先筆者先介紹 Windows Network 的歷史，在 1984 年 IBM 初進軍個人電腦網路的時代，設計一套 NetBIOS API (Network Basic Input/Output System Application Program Interface)，這組 API 只有 18 個命令來讓網路的電腦能夠建立維持和使用連接服務。後來 1985 年的時候再推出 NetBIOS 的延伸版本：NetBIOS Extended User Interface，或稱 NetBEUI，雖說 NetBEUI 是 NetBIOS 的改良版本，不過 NetBEUI 事實上可以說是一個傳輸協定，不似 NetBIOS 僅算得上是 API。

因為 NetBEUI 協定是為了小型區域網路而生的，所以首重速度，也因此有個致命的弱點，它是 non-routable 協定，也就是不能夠和其它網路（跨 router）的機器溝通。後來在 NetBIOS 介面下除了可使用 non-routable NetBEUI 協定外，亦支援 routable 的協定如 Novell IPX/SPX、TCP/IP（稱為 NetBIOS over TCP/IP，NBT）。

所以 NetBIOS 可以居於 NetBEUI、IPX/SPX 和 TCP/IP 這些協定之上。這樣有個好處，您可以改變您的通訊協定，而無需重寫您的網路服務，因為您的網路服務是針對 API 來寫的。API 會接管您的網路請求，然後運用正確的通訊協定進行工作，不過 Samba 只支援 SMB over TCP/IP。

後來微軟在 NetBIOS 之上發展出 SMB 協定，此協定最主要提供下列功能：

- 身份認證與帳號授權 (authentication and authorization of users)
- 分享檔案與印表機服務 (file and printer sharing)
- 提供 NetBIOS name 名稱解析
- 瀏覽網路資源 (Browsing)

微軟視窗作業系統使用 SMB 協定分享檔案資料及印表機資源給其他電腦，在早期的版本（如：95、98、ME、NT），SMB 使用 Port 137、138 及 139。然而，在最近版本的 Windows（如：2000、XP），SMB 資源分享則使用 445 Port。137、138、139 Port 所負責的工作請見表 12-1，NetBIOS over TCP/IP 與 OSI 7 層的對應請參考圖 1。

Port 137	Name service
Port 138	Datagram service
Port 139	Session service

表 1：SMB 協定相關 Port 列表

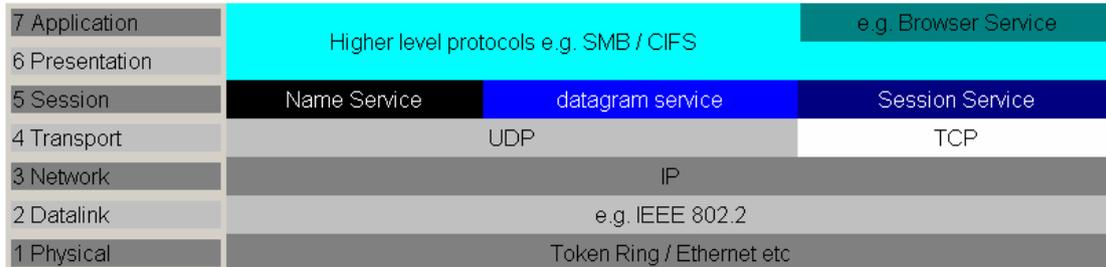


圖 1：NetBIOS over TCP/IP

資料來源：<http://ourworld.compuserve.com/homepages/timothydevans/osi.htm>

傳統上 SMB 協定使用 UDP 137 埠 (NetBIOS name service)、UDP 138 埠 (NetBIOS datagram service) 或是 139 埠 (NetBIOS session service)。一般而言 SMB 是依下列的順序建立連接 (session) 的：

- TCP Connection：在 139 埠/tcp 或是 445 埠/tcp 建立三向交握的溝通模式 (Handshaking connection)。
- NetBIOS Session Request：使用下面的「Calling Names」—本地主機的 NetBIOS 名稱加上 16th 的 0x00 字元，主伺服器的 NetBIOS 名稱加上 16th 的 0x20 字元。
- SMB Negotiate Protocol：決定使用什麼的規格交談，規格如下：PC Network Program 1.0(Core)；Microsoft Networks 1.03 (Core Plus)；Lanman 1.0(LAN Manager 1.0)；Lanman 2.1(Lan Manager 2.1)；NT LM 0.12(NT LM 0.12)。
- SMB Session Startup：依據下列五種方法傳送加密或是不加密的密碼而建立連線：Null(不加密)、Cleartext(不加密)、LM 跟 NTLM、NTLM、NTLMv2。
- SMB Tree Connect：連接到共用的名稱 (譬如 [\\servername\share](#)) 或是到服務的形態 (譬如：IPC\$ named pipe)。

RHEL 4 Samba 相關套件及設定檔

Samba 的相關套件及設定檔如下：

Daemon : nmbd, smb

Daemon 類別 : System V daemon

所需套件 :

➤ samba-*.rpm

➤ samba-client*.rpm

➤ samba-common*.rpm

Script : /etc/init.d/smb

Port : 137, 138, 139, 445

設定檔 : /etc/samba/smb.conf

Log 檔 : /var/log/samba

安裝 Samba

在 RHEL 4 安裝 Samba 是件很容易的事，只要以 root 的身份登入系統，鍵入 system-config-packages (圖 2)，勾選「網頁伺服器」即可，然後按「更新」，便會提示放入適當的光碟片以完成安裝 Samba 的工作。

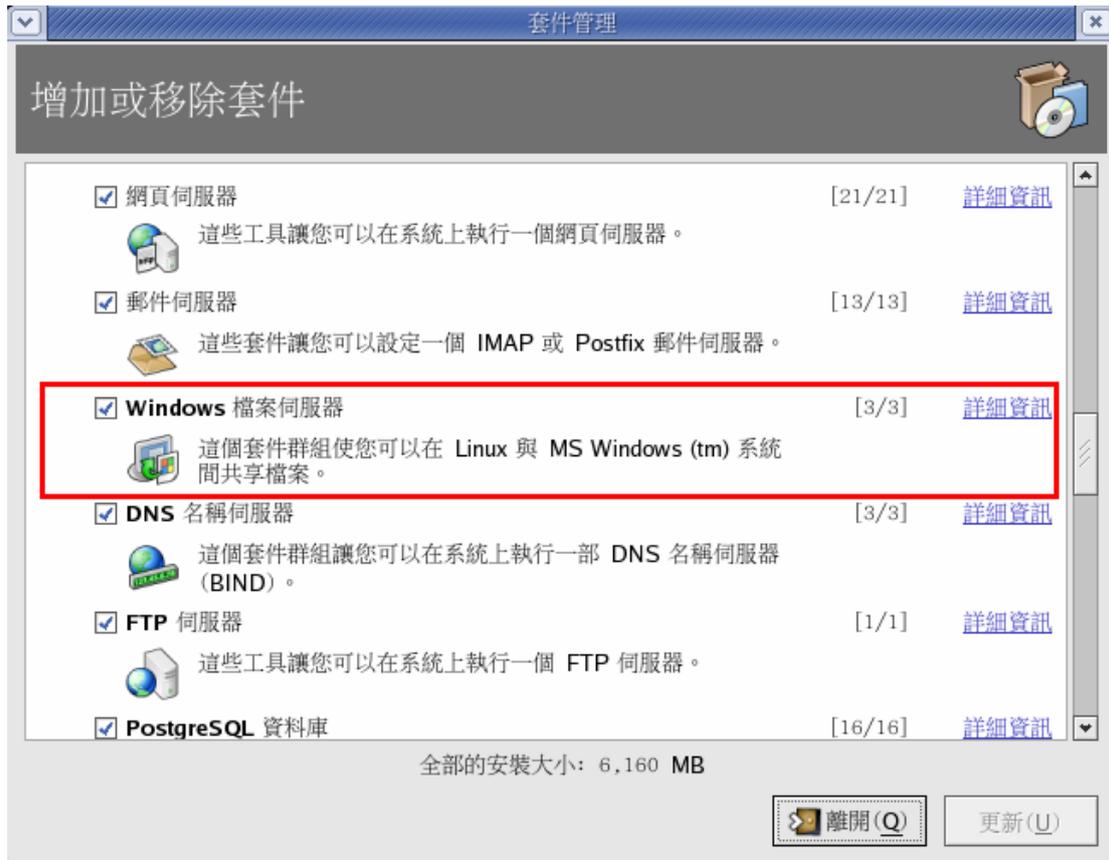


圖2：system-config-packages 畫面

待安裝後，可利用「`rpm -qa | grep '^samba'`」檢查是否安裝成功。除了採用「`system-config-packages`」工具安裝外，亦可利用 `rpm -ivh` 指令進行安裝。

```
[root@server1 ~]# rpm -qa | grep ^samba
samba-client-3.0.10-1.4E.2
samba-common-3.0.10-1.4E.2
samba-3.0.10-1.4E.2
```

Samba 設定檔解說

Samba 的設定檔有下列幾個，其中最重要的為 `/etc/samba/smb.conf`。

■ `/etc/samba/lmhosts`

這個檔案的主要目的在對應 NetBIOS name 與 IP，其設定語法與/etc/hosts 相同，只不過它是用來對應 NetBIOS 名稱與 IP，通常這個檔案可以不用設定。

```
[root@server1 /etc/samba]# cat lmhosts
```

```
127.0.0.1 localhost
```

■ /etc/samba/smbusers

這個檔案用來對應 Unix 的帳號與 Windows 的使用者名稱，語法如下：

```
Unix 帳號名稱 = Windows 的使用者名稱 Windows 的使用者名稱
```

預設值已將 Windows 的 administrator 帳號對應成 Unix 上的 root 帳號，可參考下列 smbusers 檔案內容：

```
[root@server1 /etc/samba]# cat smbusers
```

```
# Unix_name = SMB_name1 SMB_name2 ...
```

```
root = administrator admin
```

```
nobody = guest pcguest smbguest
```

■ /etc/samba/smbpasswd

筆者覺得這個檔案的重要性僅次於 smb.conf，在 Linux 上實作 Samba，有個很嚴重的管理問題，就是使用者得維護兩組密碼，一組是用來登入系統（例如：Console Login、telnet、ssh、ftp）；另一組密碼用來登入 Samba 伺服器。登入系統的密碼存放於「/etc/shadow」，而 Samba 密碼預設則是存放於「/etc/samba/smbpasswd」。

讀者可能有一個疑問，為什麼不能整合成一個？雖說 Samba 中有關登入使用者的認證方式預設為「security = user」，即登入使用者必需為本機的使用者；但因 Windows 95 OSR2 及 NT4+SP3 以後的版本在跟另一個 Windows 主機（或是 Samba Server）認證時，會將密碼經過加密後，再加以傳送，偏偏用的又不是 Linux 即有的處理方式（MD5、DES...等），所以 Samba Server 收到密碼不能直接跟/etc/shadow 比對，因為大家的密碼處理規則不同，所以只好利用

「smbpasswd 指令」，依據 Windows 規則產生密碼存於

「/etc/samba/smbpasswd」，Samba Server 會將收到的密碼與

「/etc/samba/smbpasswd」比對，檢查 Client 所輸入的密碼是否正確。因為一

開始時，並未用 `smbpasswd` 指令設定 `samba password`，所以 `/etc/samba` 目錄並未存在這個檔案。

■ `/etc/samba/smb.conf`

`/etc/samba/smb.conf` 是 Samba Server 的主要設定檔，這主要分為兩部份，分別是「`[global]`」用來設定主機功能的項目，以及針對「每個分享出去的目錄的屬性設定」，預設的 `smb.conf` 裏面有詳細的註解文字，為讓讀者清楚預設的設定檔，筆者利用正規化將註解文字先過濾掉，內容如下：

```
[root@server1 /etc/samba]# grep '^[^#;]' smb.conf

#####   [global]  用來設定主機功能的項目部份   #####

[global]
  workgroup = MYGROUP ← 即 MS 網路中的工作群組
  server string = Samba Server
  printcap name = /etc/printcap
  load printers = yes
  cups options = raw
  log file = /var/log/samba/%m.log
  max log size = 50
  security = user
  # security = user 為預設的認證方式，代表必需是 Samba Server 上的
  #本機帳號，才可進入此 Samba Server，其他相關認證方式為：
  #security = <user|share|domain|ads|server>
  #security = share，代表不需帳號／密碼即可登入 Samba Server，
  #但不要認為就可以任意存取 Samba 上的資源。還得看各個分享資源是否允
  #許此帳號可以存取。其中 domain、ads、server 通常是為了與微軟網域的
  #AD 或 PDC 整合才會#使用這些認證方式
  ...

#####   針對每個分享出去的目錄的屬性設定部份   #####

[homes]
  comment = Home Directories
```

browseable = no

writable = yes

[printers]

comment = All Printers

path = /var/spool/samba

browseable = no

guest ok = no

writable = no

printable = yes

實例演練

實作環境：

RedHat Enterprise Linux ES 4.0 Update 2

SELinux 關閉 (可利用 system-config-securitylevel 工具關閉 SELinux)

【演練一】無需輸入帳號及密碼即可使用 Samba 的/tmp 空間

步驟 1：修改 smb.conf

[global]區段部份，將工作群組改為「RHCE」，並將 smb.conf 中「security = user」改為「security = share」。找到 smb.conf 中有[tmp]分享資源的設定範例，將此區段每行前面的「;」註解符號刪除。

```
samba:~ # vi /etc/samba/smb.conf
[global]
    workgroup = RHCE ←工作群組改為全大寫的 RHCE
    security = share (security 預設值 user)
    ...
##### 針對每個分享出去的目錄的屬性設定部份 #####
[tmp] ←資源分享名稱 (share name) 為 tmp
; comment = Temporary file space
; path = /tmp ←此分享資源實為 Linux 中的/tmp 目錄
; read only = no ←允許寫入
; public = yes ←允許匿名存取
將此區段每行前面的「;」註解符號刪除
```

步驟 2：啟動 Samba 相關 Daemon

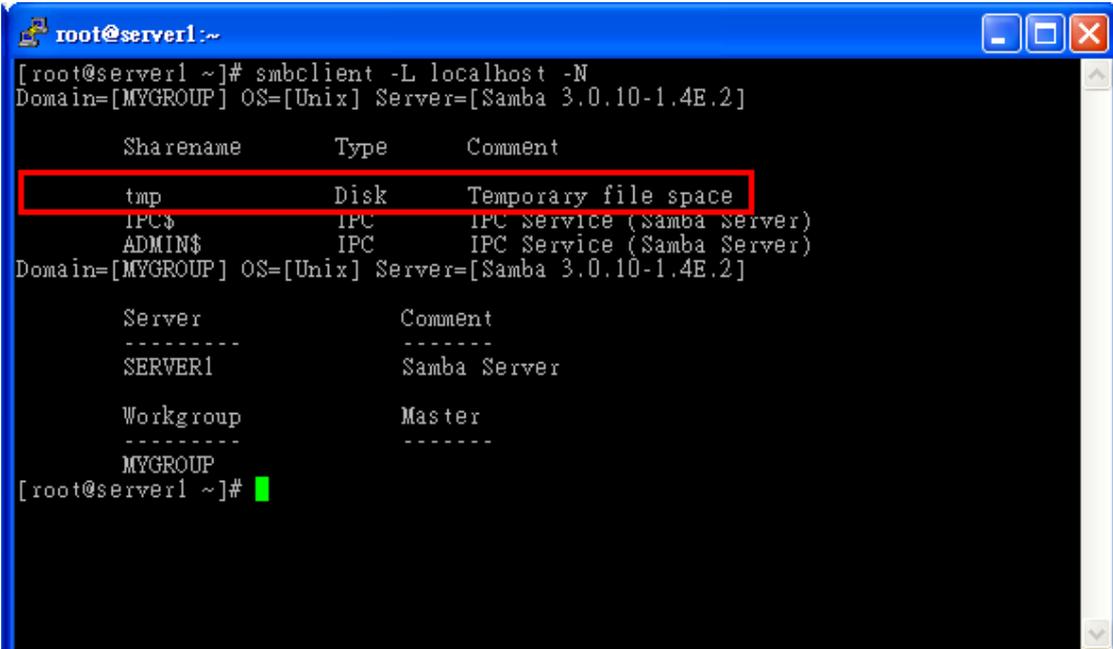
Samba 服務相關 daemon 有 smb daemon 及 nmb daemon，利用「service smb start」立即啟動這兩個 daemon；若要一開機便啟動，可利用「chkconfig smb on」指令。

```
[root@server1 ~]# service smb start
啟動 SMB 服務: [ 確定 ]
啟動 NMB 服務: [ 確定 ]
[root@server1 ~]# chkconfig smb on
```

步驟3：從Samba Server檢查是否順利分享

要檢查 Samba Server 是否已將指定目錄順利分享，可利用「smbclient -L localhost -N」指令檢查是否可看到 Samba Server 所分享出來的資源。

```
[root@server1 ~]# smbclient -L localhost -N
```



```
root@server1:~
[root@server1 ~]# smbclient -L localhost -N
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.10-1.4E.2]

  Sharename      Type            Comment
  -----
  tmp             Disk            Temporary file space
  IPC$           IPC             IPC Service (Samba Server)
  ADMIN$         IPC             IPC Service (Samba Server)
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.10-1.4E.2]

  Server                Comment
  -----
  SERVER1               Samba Server

  Workgroup              Master
  -----
  MYGROUP

[root@server1 ~]#
```

圖 3：smbclient -L localhost -N 輸出結果

步驟4：從Windows XP檢查是否可順利存取

Samba 伺服器 (一)

在Windows XP上利用「搜尋電腦」功能，輸入Samba伺服器的主機名稱或IP，點選此電腦，因為smb.conf的security設定為「share」及[tmp]分享資源中設定public = yes允許匿名存取，所以不用輸入帳號及密碼便可直接存取Samba伺服器上[tmp]分享資源。

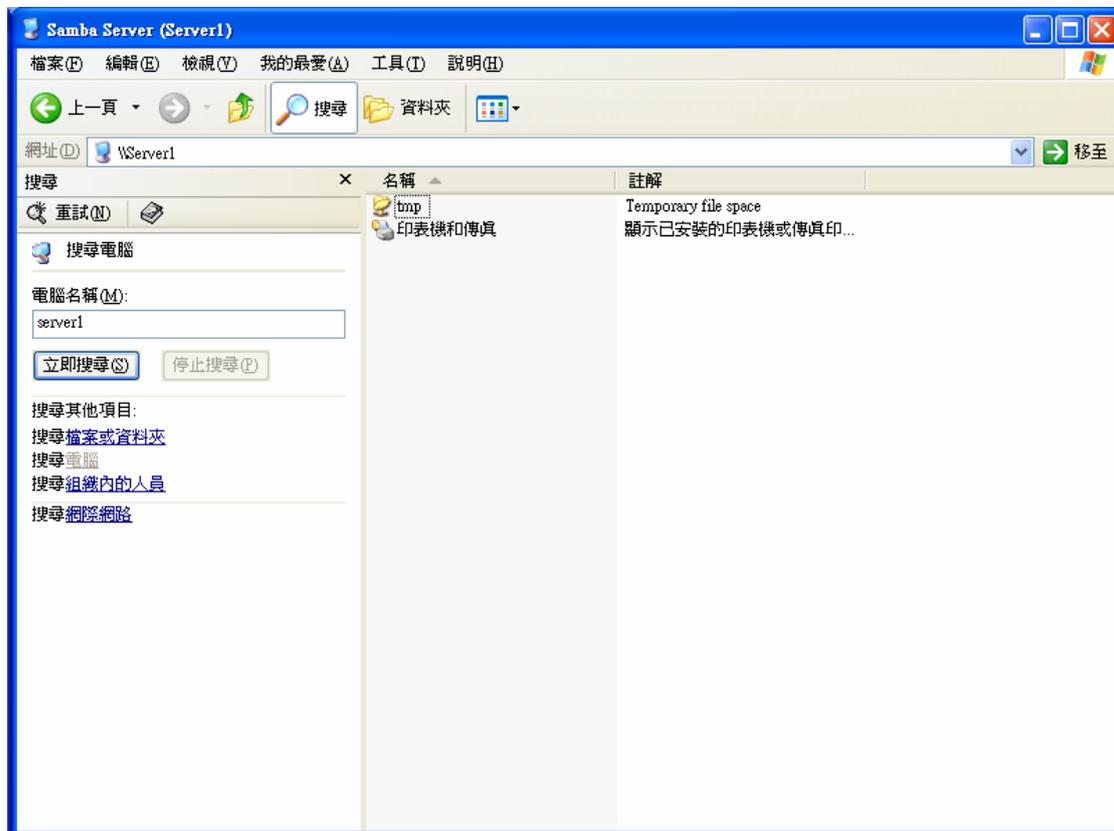


圖 4：從 Windows XP 存取 Samba 伺服器畫面

【演練二】允許某些特定帳號及才可使用 Samba 的/tmp 空間

【演練一】示範無需帳號及密碼即可存取 [tmp]分享資源，若是想要求必須需入帳號才可存取[tmp]分享資源，又該如何設定？

步驟 1：修改 smb.conf

若是讀者希望只要是 Samba Server 上本機帳號皆可存取 Samba 分享資源，必須將 security=share 改為 security=user。別忘了！還得用「smbpasswd -a 使用者名稱」幫使用者設定 Samba 密碼。

若是你只想某幾個特定的使用者才可存取 [tmp] 分享資源，而不是所有本機帳號皆可存取，請刪除 `public=yes`，並利用「`valid users = 使用者名稱列表`」設定來限制可存取的帳號，可參考下列範例：

```

security = user
...
[tmp] ← 資源分享名稱 (share name) 為 tmp
comment = Temporary file space
path = /tmp ← 此分享資源實為 Linux 中的 /tmp 目錄
read only = no ← 允許寫入
valid users = alex claire
public = yes ← 允許匿名存取

```

步驟 2：重新啟動 Samba 相關 Daemon

利用「`service smb restart`」重新啟動這 Samba 服務。

```

[root@server1 /etc/samba]# service smb restart
停止 SMB 服務: [ 確定 ]
停止 NMB 服務: [ 確定 ]
啟動 SMB 服務: [ 確定 ]
啟動 NMB 服務: [ 確定 ]

```

步驟 3：新增使用者的 Samba 密碼

使用者進入 Samba Server 需要額外的密碼，此密碼需用「`smbpasswd -a 使用者名稱`」產生且預設存於 `/etc/samba/smbpasswd` 中。

```

[root@server1 ~]# smbpasswd -a alex
New SMB password:
Retype new SMB password:
[root@server1 ~]# smbpasswd -a claire
New SMB password:

```

Retype new SMB password:

```
[root@server1 ~]# cat /etc/samba/smbpasswd
```

```
alex:500:B757BF5C0D87772FAAD3B435B51404EE:7CE21F17C0AEE7FB9  
CEBA532D0546AD6:[U          ]:LCT-43F6E32D:
```

```
claire:501:B757BF5C0D87772FAAD3B435B51404EE:7CE21F17C0AEE7FB  
9CEBA532D0546AD6:[U          ]:LCT-43F6E333:
```

步驟 4：測試

不同於【演練一】，此時從 Windows XP 連線 Samba Server，則會出現圖的畫面要求輸入帳號及密碼，輸入系統使用者的帳號名稱及 samba 密碼，便可看到分享的資源，但只有 alex 及 claire 可以存取[tmp]分享資源。



圖 5：輸入網路密碼視窗

【演練三】mis 部門資料夾

「security = share」認證方式的缺點，就是不管是否為合法的使用者，輕易就可看到 Samba Server 上所分享的資源，所以 Samba Server 預設的認證方式為 user，要登入 Samba Server 就必須是本機使用者，也就是本機使用者才能看到此台 Samba Server 到底分享多少資源，而各個分享的資源可再利用「valid users =使用者名稱列表」設定值決定那些帳號可以存取資源。

演練三將使用「security = user」的認證方式，需求為：「建立/misdata 目錄，資源分享名稱為 mis，此目錄的用途為 mis 部門 (group) 資料夾，允許本機帳號可以讀取資料，但 mis 群組可寫入資料」。

步驟 1：建立相關使用者

筆者建立下列使用者，指定其主要群組為 mis (-g mis)，且不允許這些帳號登入系統 (-s /bin/false)，只開放可從 Samba Server 登入，並記得為這些帳號設定 Samba 密碼。

```
[root@server1 ~]# groupadd mis
[root@server1 ~]# useradd -g mis -m bryan -s /bin/false
[root@server1 ~]# useradd -g mis -m eric -s /bin/false
[root@server1 ~]# useradd -g mis -m paul -s /bin/false
[root@server1 ~]# smbpasswd -a bryan
[root@server1 ~]# smbpasswd -a eric
[root@server1 ~]# smbpasswd -a paul
```

步驟 2：建立/misdata 目錄並給予適當權限

建立/misdata 目錄，將其群組修改為 mis，並開放給 group 有 write 的權限。

```
[root@server1 ~]# mkdir /misdata
[root@server1 ~]# chgrp mis /misdata
[root@server1 ~]# chmod g+w /misdata/
```

步驟 3：修改 smb.conf

並新增[mis]分享資源的設定，內容如下：

```
[mis]
  path = /misdata ← 此分享資源實為 Linux 中的/misdata 目錄
  writable list = @mis ← 允許 mis 群組寫入資料
```

步驟 4：重新啟動 Samba 相關 Daemon

利用「service smb restart」重新啟動這兩個 daemon。

```
[root@server1 /etc/samba]# service smb restart
```

```
停止 SMB 服務: [ 確定 ]
```

```
停止 NMB 服務: [ 確定 ]
```

```
啟動 SMB 服務: [ 確定 ]
```

```
啟動 NMB 服務: [ 確定 ]
```

步驟 5：測試

重新啟動 Samba 相關 daemon 後，利用 mis 群組帳號可寫入資料[mis]分享資源，但其他的本機帳號只可讀取卻無法寫入。