SSH 伺服器

SSH 是設計用來取代遠端登錄如 rlogin、rcp 和 Telnet。我們也可以使用 SSH 來編碼 X 伺服器。Public key 通常用來加密資料，而 private key 則是用來解密它。每一個使用者或主機都有它們自己的 public key 和 private key。每一個使用者想用 SSH 來連接遠端的帳號，需先建立 Public key 和 private key。然後送出它的 public key 到遠端的帳號。當這使用者想要進入這帳號時，這帳號會使用這使用者的 public key 來授權。OpenSSH 是一種免費且開放原始碼的 SSH 通訊協定的實作(Secure SHell)。 它使用安全且加密的網路連線工具來取代 telnet, ftp、rlogin、rsh 與 rcp。預設的通訊協定是預設使用 RSA 金鑰的版本 2。SSH 使用很好的加密方法，而 SSH 的網站是 www.ssh.com。

1-1ssh 伺服器

我們使用 rpm 查詢 ssh 所安裝的套件。Openssh-3.5p1-*是 ssh 伺服器在 client 端和伺服器端的核心檔。Openssh-clients-*是 openssh 使用者端連接到伺服器端的套件。Openssh-server-*是 ssh 伺服器的套件。Openssh-askpass-*是支援圖形化界面使用 SSH。Open-askpass-gnome-*是支援在 GNOME 中管理。

```
[root@flash chaiyen]# rpm -qa|grep ssh
openssh-askpass-3.5p1-6
openssh-askpass-gnome-3.5p1-6
openssh-clients-3.5p1-6
openssh-3.5p1-6
openssh-server-3.5p1-6
```

SSH 的主要組態檔在/etc/ssh/sshd_config。

#vi /etc/ssh/sshd_config

```
#       $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::
```

1-1-1 **連接埠與金鑰的位置**

(1)Port

指定 SSH **連接埠的編號，預設是** 22。

(2)ListenAddress

指定 SSHD **伺服器連接界面的** IP **位置**。

(3)HostKey

指定包含私有主機密鑰的檔案，預設是/etc/ssh_host_key。

(4)KeyrangerationInterval

伺服器密鑰將在這時間內自動產生。重新產生密鑰的因素是加強伺服器安全，讓它免於被駭客破解的風險。這個密鑰不會被儲存。假如 keyrangerationInterval **為** 0**，則密鑰不會被產生，預設是** 36000 **秒**。

(5)ServerKeyBit

定義在伺服器上密鑰的位元數量。最小的長度為 512**，而預設的長度是** 768。

1-1-2 **記錄檔與授權**

(6)SyslogFacility

設定記錄 SSHD **訊息的功能碼。這指定的功能碼有** DAEMON、USER、AUTH、LOCAL0、LOCAL1、LOCAL2、LOCAL3、LOCAL4、LOCAL5、LOCAL6 **和** LOCAL7。**預設的為** DAEMON。

(7)LoginGraceTime

假如使用者沒有成功的登錄，則在 LoginGraceTime **所指定的秒之後，伺服器就會中斷連接。假如這個數值是** 0**，則沒有時間的限制。預設是** 600 **秒**。

(8)PermitRootLogin

PermitRootLogin 指定是否超級使用者 root 可以使用 ssh 來登錄。可以設定為”yes”、”nopwd”和”no”。預設的是”yes”，允許超級使用者經過任何的授權型態登錄。”nopwd”取消超級使用者的密碼授權。”no”取消超級使用者的登錄。

(9)Strictmodes

在接受登錄之前，Strictmodes 指定是否 ssh 應該檢查這使用者家目錄與 rhosts 檔案的權限和擁用者。預設是”yes”。

(10)RSAAuthentication

指定是否允許 RSA 授權，預設是”yes”。

(11) RhostsAuthentication

指定是否授權使用 rhosts 或/etc/hosts.eauivfiles。以安全性而言，這個方法是不會被允許，一般都是使用 RhostsRSAAuthentication。預設 RhostsAuthentication 是”no”。

(12) IgnoreRhosts

指定 rhosts 和 shots 檔案將不會被使用在授權。/etc/hosts.equivand 和 /etc/shosts.equiv 繼續被使用。預設是”no”。

(13) RhostsRSAAuthentication

指定是否 rhosts 或/etc/hosts.equiv 授權成功的使用 RSA 主機授權是否允許。預設是”yes”。

(14) PasswordAuthentication

指定密碼授權是否允許。預設是”yes”。

PasswordAuthentication yes

(15) PermitEmptyPasswords

當密碼授權被允許，它指定是否這伺服器允許登錄來記錄這空的密碼字串。預設是”yes”。

(16)KerberosAuthentication
指定是否 Kerberos V5 **授權被允許。預設是** yes。

(17) KerberosOrLocalPasswd
假如 Kerberos **授權失敗，則會使用**/etc/passwd **或** SecurID **等** local **機制。預設是** no。

(18) KerberosTgtPassing
指定是否一個 Kerberos V5 TGT **可以轉送到伺服器。預設是** yes。

(19) KeepAlive
指定系統是否應該送出 keepalive **訊息到其它一邊。假如資訊有被傳送，則連接失敗或斷線都可以被發覺。預設是** yes，**因此這伺服器可以發覺是否這網路斷線或者使用者端重新開機。**
KeepAlive yes
(20) Umask
設定 sshd **預定的遮罩** umask。**遮罩是以** 0 **為開始。預設是不設遮罩。**

(21) PrintMotd
指定是否 sshd **應該列印**/etc/motd，**當一個使用者以對話方式登錄時。預設是** yes。
PrintMotd yes
(22) X11Forwarding
指定 X11 **轉送是否允許。預設是**"yes"。**取消** X11 **轉送沒有改善安全性。**
X11Forwarding yes
(23) X11DisplayOffset
指定第一個可以被 sshd **的** X11 **轉送的顯示數字。這可以預防** sshd **從真實的** X11 **伺服器介面。**

(24)PidFile
指定放置 sshd **伺服器常駐行程的行程編號** process ID。

(25)DenyShosts
這個關鍵字可以指定.shosts、.rhosts **和**/etc/hosts.equiv，**這可以禁止登錄的主機。**

1-2ssh 應用程式

這是 SSH 的應用工具。

應用程式	說明
ssh	SSH 使用者端
sshd	SSH 伺服器端常駐行程
sftp	SSH FTP 使用者端
sftp-server	SSH FTP 伺服器
scp	SSH 的使用者端 copy 指令
ssh-keygen	產生密鑰的工具
ssh-keyscan	自動聚集大眾主機密鑰，並且產生 ssh_known_hosts 檔案。
ssh-add	授權代理的增加識別。
ssh-agent	SSH 授權代理
ssh-askpass	查詢密碼的 X 視窗工具。
ssh-askpass-gnome	查詢密碼的 Gnome 工具。
ssh-signer	簽署主機基礎授權封包。
slogin	遠端登錄。

在 RedHat Linux Fedora 作業系統中，我們可以用 service 指令來啟動、重新啟動和停止 sshd 伺服器。

#service sshd start

#service sshd restart

#service sshd stop

1-2-1SSH 登錄過程

當一個使用者成功登錄時， sshd **會有下列過程。**

1. **假如以** tty **登錄，而且沒有其它指令，則會列印出最後登錄時間和**/etc/motd。
2. **假如是以** tty **登錄，則記錄登錄時間。**
3. **檢查**/etc/nologin**，顯示內容然後離開。**
4. **以一般使用者權限執行。**
5. **設定基本環境。**
6. **讀取**/etc/environment。
7. **讀取**$HOME/.ssh/environment。
8. **改變到使用者家目錄。**
9. **假如**$HOME/.ssh/rc **存在，執行使用者的** shell；**假如**/etc/sshrc **存在，執行** /bin/sh；**否則執行** xauth。Rc **檔案給予** X11 **授權協定。**
10. **執行使用者的** shell **或指令。**

1-3 設定 OpenSSH 用戶端

如要從一部用戶端機器連線到一部 OpenSSH **伺服器，用戶端機器必須已安裝** openssh-clients **與** openssh **套件。**

1-3-1SSH 伺服器的登錄

ssh 指令是 rlogin、rsh 以及 telnet 等指令之強調安全性的替代品。 它使我們可以登入到遠端的機器以及在遠端機器上執行指令。我們使用 ssh aasir.com 來登錄 aasir.com 的網站。

```
[root@flash chaiyen]# ssh aasir.com
root@aasir.com's password:
Last login: Wed Sep 24 15:52:54 2003
```

我們使用 ssh -keygen 指令來建立 public key 和 private key。我們可以指定使用何種加密方式。我們可以使用 DSA 或 RSA 加密方式。我們使用-t 選項和加密方式(dsa 或 rsa)。在這裏我們使用 rsa 的方式來加密。

#ssh –keygen –t rsa

這 ssh 登錄操作就像 rlogin 指令一樣。我們輸入 ssh 指令，然後再輸入遠端主機的位址，使用-l 的選項。

#ssh aasir.com –l chaiyen

```
[root@flash chaiyen]# ssh aasir.com -l chaiyen
chaiyen@aasir.com's password:
[chaiyen@aasir chaiyen]$
```

我們可以使用網站位址加上使用者名稱 chaiyen 來登錄 ssh。

#ssh chaiyen@aasir.com

```
[root@flash chaiyen]# ssh chaiyen@aasir.com
chaiyen@aasir.com's password:
[chaiyen@aasir chaiyen]$ _
```

1-3-2 使用 scp

我們可以使用 scp 指令來透過一個安全且加密的連線在機器間傳輸檔案，就如同 rcp 指令。
傳輸一個本機檔案到遠端機器的一般語法如下：

scp localfile username@tohostname:/newfilename

localfile 代表來源檔案，而 username@tohostname:/newfilename 代表目的地。

如要傳輸本機檔案所有的 tif 檔圖型到我們在 aasir.com 主機中的帳號，請在 shell 提示符號下輸入 scp *.tif chaiyen@aasir.com:/home/chaiyen。這是將所有圖檔的資料拷到 aasir.com 網站下的/home/chaiyen 目錄去。

```
[root@flash chaiyen]# scp *.tif chaiyen@aasir.com:/home/chaiyen
chaiyen@aasir.com's password:
002.tif                   100% |*****************************|  1127 KB    00:01
003.tif                   100% |*****************************|  1146 KB    00:01
005.tif                   100% |*****************************|  1146 KB    00:01
```

1-3-3 使用 sftp 指令
sftp 工具可以使用來開啟一個安全的且互動式的 FTP 連線，它類似 ftp，不過它是使用一種安全且加密的連線。一般的語法是 sftp username@hostname.com。一旦認證通過後，我們就可以使用類似 FTP 中的指令。我們使用 sftp chaiyen@aasir.com 來登錄 aasir.com，並且輸入 chaiyen 使用者的密碼。進入網站後，我們使用 ls 就可以觀看目錄的內容。

```
[root@flash chaiyen]# sftp chaiyen@aasir.com
Connecting to aasir.com...
chaiyen@aasir.com's password:
sftp> ls
.
..
.bash_history
.bash_logout
.bash_profile
.bashrc
```

1-3-4 產生金鑰

假如我們不想要每次使用 ssh、scp 或 sftp 來連線到遠端機器時都必須輸入密碼，我們可以產生一個認證的金鑰。必須產生金鑰給每一個使用者，我們以想要連線到遠端機器的使用者身份執行下列的步驟。假如我們以超級使用者完成以下步驟，只有超級使用者才能使用這些金鑰。SSH 通訊協定使用 ~/.ssh/authorized_keys、~/.ssh/known_hosts 與 /etc/ssh/ssh_known_hosts 檔案。預設使用 SSH 通訊協定 2 以及 RSA 金鑰。

如果要產生一個 RSA 金鑰，請在 shell 提示符號下輸入以下指令：

#ssh-keygen -t rsa

請接受 ~/.ssh/id_rsa 的預設檔案位置，並輸入一個與我們本機帳號的密碼不同的通行密碼 (passphrase)，然後再輸入一次以作確認。公鑰將會寫入到 ~/.ssh/id_rsa.pub，而私鑰則是寫入到 ~/.ssh/id_rsa。私鑰是只有我們自己才有。

然後使用 chmod 755 ~/.ssh 指令來變更 .ssh 目錄的權限設定。

#chmod 755 ~/.ssh

複製 ~/.ssh/id_rsa.pub 的內容到我們想要連線之機器的 ~/.ssh/authorized_keys 檔案中，假如 ~/.ssh/authorized_keys 不存在，我們可以複製 ~/.ssh/id_rsa.pub 檔案到另一部機器的 ~/.ssh/authorized_keys 檔案。

1-3-5 設定 ssh-agent

ssh-agent 可以使用來儲存我們的通行密碼，所以我們不需要再每次建立 ssh 或 scp 連線時輸入密碼。

在 shell 提示符號下輸入以下指令：

#exec /usr/bin/ssh-agent $SHELL

然後再輸入 ssh-add 指令。

#ssh-add

然後輸入我們的通行密碼。假如我們有設定一個以上的金鑰，我們將會收到每一個提示要求輸入密碼。當我們登出後，我們的通行密碼將會被釋放，當我們每次登入到一個虛擬主控台或開啟一個終端機視窗，我們必須執行這兩個指令。