<https://www.cnblogs.com/ggjucheng/archive/2012/08/19/2646027.html>

## ****前言****

简单的来说，SSH 是 **S**ecure **SH**ell protocol 的简写 (安全的壳程序协议)，它可以透过数据封包加密技术，将等待传输的封包加密后再传输到网络上， 因此，数据讯息当然就比较安全啰！这个 SSH 可以用来取代较不安全的 finger, R Shell (rcp, rlogin, rsh 等), talk 及 telnet 等联机模式。

## ****启动ssh服务器****

事实上，在我们使用的 Linux 系统当中，默认就已经含有 SSH 的所有需要的软件了！这包含了可以产生密码等协议的[OpenSSL](http://www.openssl.org/" \t "_blank) 软件与 [OpenSSH](http://www.openssh.com/" \t "_blank) 软件 ([注1](http://vbird.dic.ksu.edu.tw/linux_server/0310telnetssh_2.php#ps1))，所以呢，要启动 SSH 真的是太简单了！就直接给他启动就是了！此外，在目前的 Linux Distributions 当中，都是预设启动 SSH 的，所以一点都不麻烦，因为不用去设定，他就已经启动了！ 哇！真是爽快～无论如何，我们还是得说一说这个启动的方式吧！直接启动就是以 SSH daemon ，简称为 sshd 来启动的，所以，手动可以这样启动：

/etc/init.d/ssh start

需要注意的是，SSH 不但提供了 shell 给我们使用，亦即是 ssh protocol 的主要目的，同时亦提供了一个较为安全的 FTP server ，亦即是 ssh-ftp server 给我们当成是 FTP 来使用！所以，这个 sshd 可以同时提供 shell 与 ftp 喔！而且都是架构在 port 22 上面的呢！

## ****ssh客户端****

ssh命令行格式

[复制代码](javascript:void(0);)

ssh [-f] [-o 参数项目] [-p 非正规埠口] [账号@]IP [指令]

选项与参数：

-f ：需要配合后面的 [指令] ，不登入远程主机直接发送一个指令过去而已；

-o 参数项目：主要的参数项目有：

ConnectTimeout=秒数 ：联机等待的秒数，减少等待的时间

StrictHostKeyChecking=[yes|no|ask]：预设是 ask，若要让 public key

主动加入 known\_hosts ，则可以设定为 no 即可。

-p ：如果你的 sshd 服务启动在非正规的埠口 (22)，需使用此项目；

[指令] ：不登入远程主机，直接发送指令过去。但与 -f 意义不太相同。

[复制代码](javascript:void(0);)

一般使用 ssh 登入远程主机，都会填写【ssh 账号@主机IP 】的格式， 意思是说，使用该主机的某账号登入的意思。但是很多朋友都不喜欢写账号，亦即使用【ssh  主机IP 】的格式

直接联机登入到对方主机的方法 (以登入本机为例)

[复制代码](javascript:void(0);)

[root@www ~]# ssh 127.0.0.1

The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.

RSA key fingerprint is eb:12:07:84:b9:3b:3f:e4:ad:ba:f1:85:41:fc:18:3b.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '127.0.0.1' (RSA) to the list of known hosts.

root@127.0.0.1's password: <==在这里输入 root 的密码即可！

Last login: Mon Jul 25 11:36:06 2011 from 192.168.1.101

[root@www ~]# exit <==离开这次的 ssh 联机

# 由于 ssh 后面没有加上账号，因此预设使用当前的账号来登入远程服务器

[复制代码](javascript:void(0);)

如同上面的范例情况。要注意喔，如果不写账号的话，那么会以本地端计算机的账号来尝试登入远程。 也就是说，如果近端与远程具有相同的账号，那么不写账号也没有关系，如上表中的范例。但是，为了以后习惯着想， 还是一开始就使用类似 email 的方式来登入远程主机，这样的行为习惯比较好啦！

上面出现的讯息中，开头 RSA 的那行后面接的就是远程服务器的公钥指纹码，如果确定该指纹码没有问题，那么你就得要输入 yes 来将该指纹码写入服务器公钥记录文件 (~/.ssh/known\_hosts)，以方便未来比对该服务器的正确性之用。 注意是要写 yes 喔，单纯输入 Y 或 y 是不会被接受的～此外， 由于该主机的公钥已经被记录，因此未来重复使用 ssh 登入此主机时，就不会出现这个指纹码提示了。

使用 student 账号登入本机

[root@www ~]# ssh student@127.0.0.1

student@127.0.0.1's password:

[student@www ~]$ exit

# 由于加入账号，因此切换身份成为 student 了！另外，因为 127.0.0.1 曾登入过，

# 所以就不会再出现提示你要增加主机公钥的讯息啰！

登入对方主机执行命令后退出

[复制代码](javascript:void(0);)

[root@www ~]# ssh student@127.0.0.1 find / &> ~/find1.log

student@localhost's password:

# 此时你会发现怎么画面卡住了？这是因为上头的指令会造成，你已经登入远程主机，

# 但是执行的指令尚未跑完，因此你会在等待当中。那如何指定系统自己跑？

与上题相同，但是让对方主机自己跑该指令，你立刻回到近端主机继续工作：

[root@www ~]# ssh -f student@127.0.0.1 find / &> ~/find1.log

# 此时你会立刻注销 127.0.0.1 ，但 find 指令会自己在远程服务器跑喔！

[复制代码](javascript:void(0);)

如果你想要让远程主机进行关机的指令，如果不加上 -f 的参数， 那你会等待对方主机关机完毕再将你踢出联机，这比较不合理。因此，加上 -f 就很重要～因为你会指定远程主机自己跑关机， 而不需要在空空等待。例如：『ssh -f root@some\_IP shutdown -h now 』之类的指令啰。

删除掉 known\_hosts 后，重新使用 root 联机到本机，且自动加上公钥记录

[root@www ~]# rm ~/.ssh/known\_hosts

[root@www ~]# ssh -o StrictHostKeyChecking=no root@localhost

Warning: Permanently added 'localhost' (RSA) to the list of known hosts.

root@localhost's password:

# 如上所示，不会问你 yes 或 no 啦！直接会写入 ~/.ssh/known\_hosts 当中！

 此时如果每台计算机都在主动加上公钥文件记录，都得要输入『 yes 』，会累死！那么加上这个 StrictHostKeyChecking=no 就很有帮助啦！他会不询问自动加入主机的公钥到档案中，对于一般使用者帮助不大，对于程序脚本来说， 这玩意儿可就很不错用了！

 转自 <http://vbird.dic.ksu.edu.tw/linux_server/0310telnetssh_2.php#ssh_start>