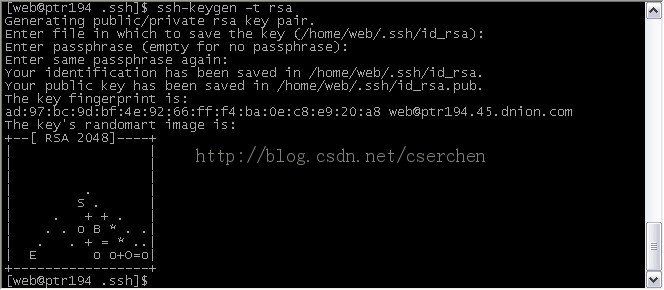
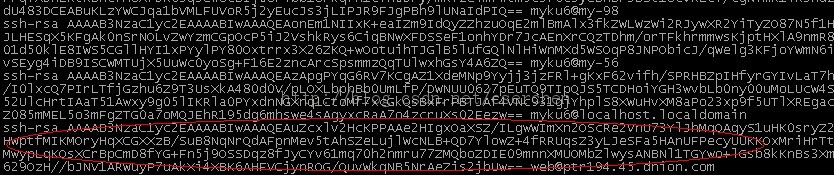
[**Linux服务器间信任关系建立方法**](http://blog.csdn.net/cserchen/article/details/18556493)

分类： [Linux](http://blog.csdn.net/cserchen/article/category/662800) 2014-01-20 15:57 134人阅读 [评论](http://blog.csdn.net/cserchen/article/details/18556493#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/cserchen/article/details/18556493#report)

**Linux两台服务器间建立信任关系的方法**  
  
  
  
在Linux服务器之间建立信任关系，是很多线上服务系统的基础性工作，这样能便于程序在多台服务器之间自动传输数据，或者方便用户不输入密码就可以在不同的主机间完成登录或者各种操作。  
  
  
  
网上关于建立Linux信任关系（ssh trust）的中文文章有一些，但是写得都不太详细，这里汇总了方方面面的资料，把多机信任关系建立方法说说清楚（文/陈运文）  
  
  
  
一 建立信任关系的基本操作  
  
  
  
基本场景是想从一台Server服务器直接登录另一台，或者将Server服务器的数据不需密码验证直接拷贝至Client服务器，以下我们简称Server服务器为S（待发送的数据文件在这台服务器上），Client服务为C，信任关系的最简单操作方法如下：  
  
  
  
**1 在S服务器上，进入当前用户根目录下的隐藏目录 .ssh，命令如下**：  
  
cd ~/.ssh  
  
（注：目录名前的点好”.”表示该文件夹是一个特殊的隐藏文件夹，ls命令下默认是看不到的，通过 ls –a 命令观察到）  
  
  
**2 生成S服务器的私钥和公钥：**  
  
ssh-keygen -t rsa  
  
（注：rsa是一种加密算法的名称，此处也可以使用dsa，关于rsa和dsa算法的介绍可见本文后半章节）  
  
  
  
  
  
  
ssh-keygen生成密钥用于信任关系生成  
  
  
  
-此时会显示Generating public/private key pair. 并提示生成的公钥私钥文件的存放路径和文件名，默认是放在 /home/username/.ssh/id\_rsa 这样的文件里的，通常不用改，回车就可以  
  
然后Enter passphrase(empty for no passphrase): 通常直接回车，默认不需要口令  
  
Enter same passphrase again: 也直接回车  
  
然后会显式密钥fingerprint生成好的提示，并给出一个RSA加密协议的方框图形。此时在.ssh目录下ls，就可以看到生成好的私钥文件id\_rsa和公钥文件id\_rsa.pub了  
  
  
  
**以下是各种补充说明：**  
  
注1：如果此时提示 id\_rsaalready exists，Overwrite(y/n) 则说明之前已经有人建好了密钥，此时选择n 忽略本次操作就行，可以直接用之前生成好的文件；当然选y覆盖一下也无妨  
  
  
注2：公钥用于加密，它是向所有人公开的（pub是公开的单词public的缩写）；私钥用于解密，只有密文的接收者持有。  
  
**3 在Server服务器上加载私钥文件**  
  
仍然在.ssh目录下，执行命令：

ssh-add id\_rsa  
  
系统如果提示：Identity added: id\_rsa (id\_rsa) 就表明加载成功了  
  
  
下面有几个异常情况处理：  
  
  
–如果系统提示：could not open a connection to your authentication agent  
  
则需要执行一下命令：  
  
ssh-agent bash  
  
然后再执行上述的ssh-add id\_rsa命令  
  
  
–如果系统提示id\_rsa: No such file or directory  
  
这是系统无法找到私钥文件id\_rsa，需要看看当前路径是不是不在.ssh目录，或者私钥文件改了名字，例如如果建立的时候改成 aa\_rsa，则这边命令中也需要相应改一下  
  
-如果系统提示 command not found，那肯定是你命令敲错字符了J  
  
-提示Agent admitted failure to sign using the key，私钥没有加载成功，重试ssh-add  
  
-注意id\_rsa/id\_rsa.pub文件不要删除，存放在.ssh目录下  
  
  
  
**4 把公钥拷贝至Client服务器上**  
  
很简单，例如 scp id\_rsa.pub user@10.11.xx.xx:~  
  
  
  
**5 ssh登录到Client服务器上，然后在Client服务器上，把公钥的内容追加到authorized\_keys文件末尾**（这个文件也在隐藏文件夹.ssh下，没有的话可以建立，没有关系）  
  
cat id\_rsa.pub >> ~/.ssh/authorized\_keys  
  
  
**以下是各种补充说明，遇到问题时可以参考**：  
  
注1：这里不推荐用文件覆盖的方式，有些教程直接scp id\_rsa.pub 到Client服务器的authorized\_keys文件，会导致之前建的其他信任关系的数据被破坏，追加到末尾是更稳妥的方式；  
  
注2： cat 完以后，Client服务器上刚才拷贝过来的id\_rsa.pub文件就不需要了，可以删除或移动到其它地方）  
  
注3：ssh-keygen 命令通过-b参数可以指定生成的密钥文件的长度，如果不指定则默认为1024，如果ssh-keygen –b 4096（最长4096），则加密程度提高，但是生成和验证时间会增加。对一般的应用来说，默认长度已经足够胜任了。如果是rsa加密方式，那么最短长度 为768 byte  
  
注4：authorized\_keys文件的权限问题。如果按上述步骤建立关系后，仍然要验证密码，并且没有其他报错，那么需要检查一下authorized\_keys文件的权限，需要作下修改： chmod g-w authorized\_keys  
  
  
OK，现在试试在Server端拷贝一个文件到Client服务器，应该无需交互直接就传过去了。  
  
  
  
但是此时从Client传数据到Server服务器，仍然是需要密码验证的。如果需要两台服务器间能直接互传数据，则反过来按上述步骤操作一下就可以了  
  
  
  
二 删除服务器间信任关系的方法  
  
  
  
如果想取消两台服务器之间的信任关系，直接删除公钥或私钥是没有用的，需要在Client服务器上，打开 ~/.ssh/ authorized\_keys 文件，找到对应的服务器的公钥字段并删除  
  
  
  
每个段落的开头是ssh-rsa字样，段尾是Server服务器的帐号和ip（如下图红框），需要细心的找一下后删除整段  
  
  
  
  
密钥文件内容和删除Linux服务器间信任关系的方法  
  
  
  
  
  
三 各种可能遇到的情况和处理方法  
  
–提示 port 22: Connection refused  
  
可能的原因：没有正确安装最新的openssh-server，安装方法如下  
  
sudo apt-get install openssh-server  
  
不支持apt安装的，可以手工下载：  
  
wget http://ftp.ssh.com/pub/ssh/ssh-3.2.9.1.tar.gz  
  
–关于目录和文件的权限设置  
  
.ssh目录的权限必须是700，同时本机的私钥的权限必须设置成600：  
  
chmod 600 id\_rsa  
  
否则ssh服务器会拒绝登录  
  
  
  
四 关于RSA和DSA加密算法  
  
在ssh-keygen命令中，-t参数后指定的是加密算法，可以选择rsa或者dsa  
  
  
  
RSA 取名自算法的三位提出者Ron Rivest, Adi Shamir, and Leonard Adleman的姓名首字母，作为一种非对称加密算法，RSA的安全性基于及其困难的大整数分解（两个素数的乘积的还原问题）。关于RSA算法原理的文章 很多，感兴趣的朋友可以找来读一读。  
  
  
  
DSA = Digital Signature Algorithm，基于有限域离散对数难题，是Schnorr和ElGamal签名算法的变种，一般用于数字签名和认证，被美国标准局（NIST）采纳 为数字签名标准DSS（Digital Signature Standard），based on discrete logarithms computation.  
  
DES = Digital Encryption Standard. Obsolete standard.  
  
  
  
RSA算法好在网络容易实现密钥管理,便进行数字签名,算法复杂,加/解速度慢,采用非对称加密。在实际用于信任关系建立中，这两种方法的差异很微小，可以挑选其一使用。  
  
  
  
五 关于SSH协议的介绍  
  
SSH全称Secure SHell，顾名思义就是非常安全的shell的意思，SSH协议是IETF（Internet Engineering Task Force）的Network Working Group所制定的一种协议。SSH的主要目的是用来取代传统的telnet和R系列命令（rlogin,rsh,rexec等）远程登陆和远程执行命令 的工具，实现对远程登陆和远程执行命令加密。防止由于网络监听而出现的密码泄漏，对系统构成威胁。  
  
ssh协议目前有SSH1和SSH2，SSH2 协议兼容SSH1。目前实现SSH1和SSH2协议的主要软件有OpenSSH和SSH Communications Security Corporation　公司的SSH Communications 软件。前者是OpenBSD组织开发的一款免费的SSH软件，后者是商业软件，因此在linux、FreeBSD、OpenBSD、NetBSD等免费类 UNIX系统种，通畅都使用OpenSSH作为SSH协议的实现软件。因此，本文重点介绍一下OpenSSH的使用。需要注意的是OpenSSH和SSH Communications的登陆公钥/私钥的格式是不同的，如果想用SSH Communications产生的私钥/公钥对来登入到使用OpenSSH的linux系统需要对公钥/私钥进行格式转换。  
  
第一次登陆后，ssh就会把登陆的ssh指纹存 放在用户home目录的.ssh目录的know\_hosts文件中，如果远程系统重装过系统，ssh指纹已经改变，你需要把 .ssh 目录下的know\_hosts中的相应指纹删除，再登陆回答yes，方可登陆。请注意.ssh目录是开头是”.”的隐藏目录，需要ls –a参数才能看到。而且这个目录的权限必须是700,并且用户的home目录也不能给其他用户写权限，否则ssh服务器会拒绝登陆。如果发生不能登陆的问 题，请察看服务器上的日志文件/var/log/secure。通常能很快找到不能登陆的原因。  
  
  
  
六 关于ssh\_config和sshd\_config文件配置的说明  
  
/etc/ssh/ssh\_config:  
  
Host \*  
  
选项“Host”只对能够匹配后面字串的计算机有效。“\*”表示所有的计算机。  
  
ForwardAgent no  
  
“ForwardAgent”设置连接是否经过验证代理（如果存在）转发给远程计算机。  
  
ForwardX11 no  
  
“ForwardX11”设置X11连接是否被自动重定向到安全的通道和显示集（DISPLAY set）。  
  
RhostsAuthentication no  
  
“RhostsAuthentication”设置是否使用基于rhosts的安全验证。  
  
RhostsRSAAuthentication no  
  
“RhostsRSAAuthentication”设置是否使用用RSA算法的基于rhosts的安全验证。  
  
RSAAuthentication yes  
  
“RSAAuthentication”设置是否使用RSA算法进行安全验证。  
  
PasswordAuthentication yes  
  
“PasswordAuthentication”设置是否使用口令验证。  
  
FallBackToRsh no  
  
“FallBackToRsh”设置如果用ssh连接出现错误是否自动使用rsh。  
  
UseRsh no  
  
“UseRsh”设置是否在这台计算机上使用“rlogin/rsh”。  
  
BatchMode no  
  
“BatchMode”如果设为“yes”，passphrase/password（交互式输入口令）的提示将被禁止。当不能交互式输入口令的时候，这个选项对脚本文件和批处理任务十分有用。  
  
CheckHostIP yes  
  
“CheckHostIP”设置ssh是否查看连接到服务器的主机的IP地址以防止DNS欺骗。建议设置为“yes”。  
  
StrictHostKeyChecking no  
  
“StrictHostKeyChecking”如果设置成“yes”，ssh就不会自动把计算机的密匙加入“$HOME/.ssh/known\_hosts”文件，并且一旦计算机的密匙发生了变化，就拒绝连接。  
  
IdentityFile ~/.ssh/identity  
  
“IdentityFile”设置从哪个文件读取用户的RSA安全验证标识。  
  
Port 22  
  
“Port”设置连接到远程主机的端口。  
  
Cipher blowfish  
  
“Cipher”设置加密用的密码。  
  
EscapeChar ~  
  
“EscapeChar”设置escape字符。  
  
/etc/ssh/sshd\_config:  
  
  
  
  
  
Port 22  
  
“Port”设置sshd监听的端口号。  
  
ListenAddress 192.168.1.1  
  
“ListenAddress”设置sshd服务器绑定的IP地址。  
  
HostKey /etc/ssh/ssh\_host\_key  
  
“HostKey”设置包含计算机私人密匙的文件。  
  
ServerKeyBits 1024  
  
“ServerKeyBits”定义服务器密匙的位数。  
  
LoginGraceTime 600  
  
“LoginGraceTime”设置如果用户不能成功登录，在切断连接之前服务器需要等待的时间（以秒为单位）。  
  
KeyRegenerationInterval 3600  
  
“KeyRegenerationInterval”设置在多少秒之后自动重新生成服务器的密匙（如果使用密匙）。重新生成密匙是为了防止用盗用的密匙解密被截获的信息。  
  
PermitRootLogin no  
  
“PermitRootLogin”设置root能不能用ssh登录。这个选项一定不要设成“yes”。  
  
IgnoreRhosts yes  
  
“IgnoreRhosts”设置验证的时候是否使用“rhosts”和“shosts”文件。  
  
IgnoreUserKnownHosts yes  
  
“IgnoreUserKnownHosts”设置ssh daemon是否在进行RhostsRSAAuthentication安全验证的时候忽略用户的“$HOME/.ssh/known\_hosts”  
  
StrictModes yes  
  
“StrictModes”设置ssh在接收登录请求之前是否检查用户家目录和rhosts文件的权限和所有权。这通常是必要的，因为新手经常会把自己的目录和文件设成任何人都有写权限。  
  
X11Forwarding no  
  
“X11Forwarding”设置是否允许X11转发。  
  
PrintMotd yes  
  
“PrintMotd”设置sshd是否在用户登录的时候显示“/etc/motd”中的信息。  
  
SyslogFacility AUTH  
  
“SyslogFacility”设置在记录来自sshd的消息的时候，是否给出“facility code”。  
  
LogLevel INFO  
  
“LogLevel”设置记录sshd日志消息的层次。INFO是一个好的选择。查看sshd的man帮助页，已获取更多的信息。  
  
RhostsAuthentication no  
  
“RhostsAuthentication”设置只用rhosts或“/etc/hosts.equiv”进行安全验证是否已经足够了。  
  
RhostsRSAAuthentication no  
  
“RhostsRSA”设置是否允许用rhosts或“/etc/hosts.equiv”加上RSA进行安全验证。  
  
RSAAuthentication yes  
  
“RSAAuthentication”设置是否允许只有RSA安全验证。  
  
PasswordAuthentication yes  
  
“PasswordAuthentication”设置是否允许口令验证。  
  
PermitEmptyPasswords no  
  
“PermitEmptyPasswords”设置是否允许用口令为空的帐号登录。  
  
AllowUsers admin  
  
“AllowUsers”的后面可以跟着任意的数量的用户名的匹配串（patterns）或user@host这样的匹配串，这些字符串用空格隔开。主机名可以是DNS名或IP地址。