Linux中sshd服务解析

# ssh简介

第一次登陆时候，就会下载服务器上面的公钥。可以保证服务器接收到的信息不被篡改（客户端传递的数据都会用公钥加密后传递）。（公钥想象成锁，私钥想象成钥匙）

## SSH原理

　　SSH协议是建立在应用层和传输层基础上的安全协议，其主要由以下三部分组成，共同实现SSH的安全保密机制：

* 传输层协议。该协议提供诸如认证、信任和完整性检验等安全措施，此外还可以任意地提供数据压缩功能。通常情况下，这些传输层协议都建立在面向连接的TCP数据流之上。
* 用户认证协议层。用来实现服务器的跟客户端用户之间的身份认证，其运行在传输层协议之上。
* 连接协议层。分配多个加密通道至一些逻辑通道上，它运行在用户认证层协议之上。

　　SSH是由客户端和服务端的软件组成的，有两个不兼容的版本分别是：1.x和2.x。用SSH 2.x的客户程序是不能连接到SSH 1.x的服务程序上去的。OpenSSH 2.x同时支持SSH 1.x和2.x。从客户端来看，SSH提供两种级别的安全验证。

　　1) 第一种级别（基于口令的安全验证）：只要用户知道自己账号和口令，就可以登录到远程主机。所有传输的数据都会被加密，但是不能保证用户正在连接的服务器就是用户想连接的服务器。可能会有别的服务器在冒充真正的服务器，也就是受到“中间人”这种方式的攻击。

　　2) 第二种级别（基于密钥的安全验证）：需要依靠密钥，也就是用户必须为自己创建一对密钥，并把公用密钥放在需要访问的服务器上。如果用户要连接到SSH服务器上，客户端软件就会向服务器发出请求，请求用用户的密钥进行安全验证。服务器收到请求之后，先在用户在该服务器的家目录下寻找用户的公用密钥，然后把它和用户发送过来的公用密钥进行比较。如果两个密钥一致，服务器就用公用密钥加密“质询”（challenge）并把它发送给客户端软件。客户端软件收到“质询”之后就可以用用户的私人密钥解密再把它发送给服务器。用这种方式，必须知道自己密钥的口令。但是，与第一种级别相比，第二种级别不须要在网络上传送口令。第二种级别不仅加密所有传送的数据，而且“中间人”这种攻击方式也是不可能的（因为他没有用户的私人密钥）。

## 快速安装OpenSSH

（1）解压缩源码包

　tar xzvf openssh-6.1p1.tar.gz

（2）预编译，生成makefile文件

　cd openssh-6.1p1

　./configure

（3）安装

　make && make install

　安装完成了之后，可以使用下述命令（Red Hat和Fedora系列）进行启动：

　service sshd start或者/etc/initd/sshd start

# ssh配置文件

/etc/ssh/sshd\_config(服务端)

/etc/ssh/ssh\_config(客户端)

## 常见配置解析

|  |
| --- |
| Port 22 #设置sshd监听端口，为了安全，建议使用22默认端口  AddressFamily any  ListenAddress 0.0.0.0 #用来设置sshd服务器绑定端口  ListenAddress ::  Protocol 2 #ssh协议的版本  HostKey /etc/ssh/ssh\_host\_rsa\_key  HostKey /etc/ssh/ssh\_host\_dsa\_key #是指服务器密钥文件的路径  KeyRegenerationInterval 1h #多少秒后系统重新生成密钥  ServerKeyBits 1024 #定义服务器密钥长度  #用来设定在记录来自sshd的消息时候，是否给出facility code  SyslogFacility AUTHPRIV  LogLevel INFO #记录sshd日志消息的级别  #设置用户登录失败，在切断连接服务前服务器等待时间(s)  LoginGraceTime 2m  PermitRootLogin yes #是否允许root用户登录  #设置ssh在接受登录请求之前是否检查用户根目录和rhosts文件权限。  StrictModes yes  MaxAuthTries 6 #最大失败尝试次数  #MaxSessions 10  RSAAuthentication yes #是否启用RSA密钥验证  PubkeyAuthentication yes #是否开启公钥验证  AuthorizedKeysFile .ssh/authorized\_keys #设置公钥验证文件的路径  #AuthorizedKeysCommand none  #AuthorizedKeysCommandRunAs nobody  #RhostsRSAAuthentication no  #HostbasedAuthentication no  #设置进行RhostsRSAAuthentication验证时，忽略~.ssh/known\_hosts  IgnoreUserKnownHosts no  IgnoreRhosts yes #验证的时候是否使用~/.rhosts和~/.shosts  PasswordAuthentication yes #用来设置是否开启密码验证机制  #PermitEmptyPasswords no #是否允许口令为空的用户登录  #PasswordAuthentication yes  #ChallengeResponseAuthentication yes #是否启用s/key密码  UsePAM yes #是否通过PAM认证  X11Forwarding yes #是否允许x11转发  PrintMotd yes #用来设置sshd是否在用户登录时候显示/etc/motd中内容  PrintLastLog yes #是否显示上一次登录信息、  #TCPKeepAlive yes  UseDNS yes #是否使用DNS反向解析 |

# 常用ssh命令

## ssh命令

命令格式：ssh user@host

SSH的默认端口是22，也就是说，你的登录请求会送进远程主机的22端口。使用p参数，可以修改这个端口。例如： ssh -p 2222 user@host

## sftp命令

命令格式：sftp root@ip

交互命令如下（l表示本机）

* ls：远程服务器上目录列表
* cd：远程服务器目录切换
* lls:本地目录列表
* lcd：本地目录切换
* get：下载
* put：上传
* quit：退出

# 密钥对登陆

第一步：生成公钥和私钥

ssh-keygen -t rsa -P '' -f /root/.ssh/id\_rsa

第二步：将公钥放到远程主机上（表示远程主机对本机的信任）

ssh-copy-id -i /root/.ssh/id\_rsa.pub user@server

现在就可以直接访问远程主机。如果不行在远程主机的/etc/ssh/sshd\_config中将以下部分的注释去掉。

|  |
| --- |
| RSAAuthentication yes  PubkeyAuthentication yes  AuthorizedKeysFile .ssh/authorized\_keys |

## ssh-keygen命令

作用：ssh-keygen命令网络安全 ssh-keygen命令用于为“ssh”生成、管理和转换认证密钥，它支持RSA和DSA两种认证密钥。

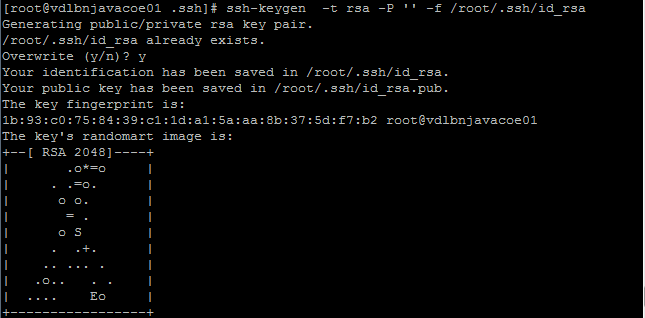
命令格式：ssh-keygen [选项]

选项：

* -b：指定密钥长度；
* -e：读取openssh的私钥或者公钥文件；
* -C：添加注释；
* -f：指定用来保存密钥的文件名；
* -i：读取未加密的ssh-v2兼容的私钥/公钥文件，然后在标准输出设备上显示openssh兼容的私钥/公钥；
* -l：显示公钥文件的指纹数据；
* -N：提供一个新密语；
* -P：提供（旧）密语；
* -q：静默模式；
* -t：指定要创建的密钥类型。可以使用："rsa1"(SSH-1) "rsa"(SSH-2) "dsa"(SSH-2)

文件说明

* id\_rsa： 私钥文件
* id\_rsa.pub // 公钥文件
* authorized\_keys // 存放客户端公钥的文件, 对应客户端可以免密登录
* known\_hosts // 确认过公钥指纹的可信服务器列表的文件，域名或者IP的ssh公钥发生过变更，会发生报错。



ssh-keygen -t rsa -P '' -f /root/.ssh/id\_rsa

## ssh-copy-id命令

ssh-copy-id命令可以把本地主机的公钥复制到远程主机的authorized\_keys文件上，ssh-copy-id命令也会给远程主机的用户主目录（home）和~/.ssh, 和~/.ssh/authorized\_keys设置合适的权限。

ssh-copy-id [-i [identity\_file]] [user@]machine

-i：指定公钥文件  
例如：ssh-copy-id -i ~/.ssh/id\_rsa.pub user@server