|  |
| --- |
| ssh配置权威指南 |
| |  | | --- | |  | |
| 来源: ChinaUnix博客 　日期： 2006.07.17 02:17　(共有条评论) [我要评论](http://linux.chinaunix.net/bbs/thread-936114-1-1.html) |
|  |
| ssh配置权威指南    Pogo Linux (Shanghai) Forums > Linux 入门及系统管理 > Ssh权威指南(转)-------------------------------------------------------------------------------- PDA 查看完整版本 : Ssh权威指南(转) -------------------------------------------------------------------------------- James 2006-01-15, 00:27 我的评价： 本书是一本介绍通信安全的书籍，如果你想保障你的通信安全，本书能给你一个很好的解决 方案。本书从ssh协议介绍起，到具体的开源实现和商业实现。但本书同时介绍开源实现和 商业实现，给人感觉比较乱。 注意：由于openssh是开源软件，各版间的配置方法和设置参数可能会不同，所以设置时要 以最新版的手册为准，不要硬套书本的设置。 第一章 SSH简介 ssh(secure shell)是一种通用，功能强大的基于软件的网络安全解决方案，计算机每次向网络 发送数据时，SSH都会自动对其进行加密。数据到达目的地时，SSH自动对加密数据进行 解密。整个过程都是透明的。它使用了现代的安全加密算法，足以胜任大型公司的任务繁重 的应用程序的要求。 SSH协议内容涉及认证AUTHENTICATION,加密ENCRYPTION,和网络上传输数据的完整 性INTEGRITY。 SSH特性概述 1、远程登录 $ ssh -l root host.example.com 2、安全文件传输 $ scp myfile  [metoo@secondacount.com](mailto:metoo@secondacount.com) 3、安全执行远程命令，能确保传输的数据安全。 4、密钥和代理 基于密钥的认证方法不用记住多个帐号密码。 5、访问控制，能授权别人访问自已帐号。 6、端口转发 $ ssh -L 3002:localhost:119 xxx.xxx.com 相关安全技术 rsh命令族，rsh,rlogin and rcp。连接不加密，认证模型十分脆弱。 PGP加密程序。它是基于文件的。 Kerberos认证系统，用于网络可能被监视，而且计算机不是中心控制的环境。基于麻省理工 学院的Athena项目。它SSH不是，SSH是轻量级的，容易部署。Kerberos使用前必须构建 一些重要基础。 IPSEC，Internet安全协议。 SRP，安全远程密码协议，是Stanford大学开发的。是一种专用的认证协议。 SSL,安全套接字。 stunnel，是一种SSL工具，它为UNIX环境现有的基于TCP的服务（POP,IMAP等）增加 SSL保护，而不用修改服务器源程序。 第二章 SSH客户端的基本用法 最常用功能，1、经由安全连接登录到远程计算机。2、通过安全连接在计算机间拷贝文件。 当你第一次连接SSH服务器时，出现问答时要回答“yes”，把已知名主机的密钥的公共部 分拷贝一份存在本地。之后你每次连接这台服务器，就用这个公钥来验证远程主机。最好是 能在首次连接前就能获得这个公钥，否则第一次连接服务器时你可能已受到中间人攻击。 使用密钥进行认证，SSH支持公钥认证，可以使用加密密钥，密钥比密码更安全。 SSH证书使用一对密钥，一个私钥，一个公钥，私钥只保存你独有的一些秘密信息。SSH 用其向服务器证明自已的身份。公钥是公开的，可以随便将其放入SSH服务器上自已的帐 号中,在认证时，进行私钥和公钥协商，如果匹配，那么身份就得以证明，认证就成功。 在使用公钥认证以前，首先要进行一些设置： 1、需要一对密钥，还需要使用一个口令来保护自已的私钥。 使用ssh-keygen程序生成一对密钥。如果不存在SSH目录，程序为自动创建本地SSH目录 ~/.ssh，并将所生成的密钥分成两个文件存储，公有部份identity.pub,私有部分identity，或 id\_dsa\_1024\_a,id\_dsa\_1024\_a.pub(ssh2)。 2、需要在SSH服务器上安装自已的公钥。 通过配置SSH目录中的一个文件实现，对于SSH1 AND OPENSSH来说该文件是 ~/.ssh/authorized\_keys。对SSH2来说是~/.ssh2/authorization。OPENSSH中的SSH-2连接自 3.0版本起也一同使用authorized\_keys文件。3.0版本前的使用authorized\_keys2。把用户本 地机器生成identity.pub文件内容加入其中。对SSH2来说，用户需编辑两个文件，一个客 户端一个在服务器端， 在客户端，要创建或编辑文件~/.ssh2/identification并在其中插入一行，说明自已的私钥文件 名： IdKey id\_dsa\_1024\_a. 在服务器端，要创建或编辑文件~/.ssh2/authorization，该文件包含有公钥信息，每行一个， 但和SSH1的authorized\_keys文件不同（authorized\_keys中包含有公钥的拷贝），authorization 中只给出公钥文件名：Key id\_dsa\_1024\_a.pub。最后，把这个文件从本地机器上拷贝到服务 器~/.ssh2中。为安全起见，要确保ssh目录的安全，只有所有者才有权写入。如果远程用户 的SSH配置文件的权限设置不当，服务器可能会拒绝进行认证。 公钥认证比密码认证更安全，因为： 1、 公钥认证需要两个加密部份（磁盘上的identify文件和用户头脑中的口令），入侵都必须 2、 具备两种条件才行。密码认证只需要一个部份，那就是密码，它可能更容易被窃取。 3、 在公钥认证中，口令和密钥都不用发给远程主机，只要把前面讨论的认证者发给远程主 4、 机就可以了，因此，并没有秘密信息传出客户端。 3、机器生成的密钥是不可能猜测出来的，而人生成的密码容易受到字典攻击。 通过禁用密码认证并只允许使用密钥认证能极大提高主机的安全性。 如果要修改密钥 如果已经生成一个密钥对，并把公钥拷贝到多个SSH服务器上了，用户有一天决定修改自 已的身份，因为再运行ssh-keygen。这样，就会覆盖identify and identify.pub文件，用户之 前的公钥就没用了，必须把新公钥再次拷贝到各个服务器上。这是很头疼的，所以建议： 1、 不能局限于仅仅使用一个密钥对，可随意生一此密钥对，将其保存在不同的文件中，并 2、 将其用作不同的用途。 3、 如果只是想修改口令，就不必重新生成一个密钥对，ssh-keygen有一个命令行选项可以 4、 替换现有的密钥口令。ssh1 and openssh是-p,对于ssh2是-e。这样，因为私钥没变，所以 5、 公钥依然不效，中需使用新口令对私钥进行解密就可以了。 ssh代理 它可以把私钥保存在内存中，为认证提供服务，不用重复输入密码。直到用户退出为止。代 理程序是ssh-agent。可手工运行也可编辑~/.login 或~/.xsession来自动运行。 $ ssh-agent $SHELL 其中SHELL是用户登录shell的环境变量。运行该命令后，打开另外 一个shell，在这个shell中可以访问代理。 接着用ssh-add命令装入私钥。这样，使用ssh and scp命令就不用再提醒输入口令了。口令 装入内存中。如果用户正运行x window系统，并设置了DISPLAY环境变量，而标准输入 不是终端，那么ssh-add就使用一个图形化X程序ssh-askpass来读取口令。要强制ssh-add 使用X来读取口令，请在命令行中输入ssh-add  代理转发 可能通过代理转发功能，可以用SCP把文件从远程SSH服务器上拷贝到第三方SSH主机上。 条件是你在第三方SSH主机上必须有一个帐号。具体操作是这样的： 1、 在本地主机上运行命令：# scp  [email=pat@shell.isp.com:print-me]pat@shell.isp.com:print-me[/email]  2、  [email=psmith@other.host.net:other-print-me]psmith@other.host.net:other-print-me[/email] 2、这个scp进程与本地代理进行联系，并让用户和shell.isp.com进行认证。 3、自动在shell.isp.com上执行第二个scp，用来把文件拷贝到other.host.net主机上。 3、 由于启用了代理转发（SSH1 AND SSH2默认是打开的，openssh默认是关闭的），因此 4、 shell.isp.com上的SSH服务器就充当一个代理。 5、第二个scp进程通过联系shell.isp.com上的代理试图让用户和other.host.net进行认证。 5、 shell.isp.net上ssh服务器秘密与用户的本地代理进行通信，从而构建出一个认证者来提 6、 供用户的证书并将其传回服务器。 7、服务器为第二个scp进程验证用户的身份，other.host.net上的认证获得成功。 8、开始拷贝文件。 由于代理转发不会把私钥发送到远程主机上，而是把认证返回第一台机进行处理，因此密钥 是安全的。 不用密码或口令进行连接方式 1、使用代理的公钥认证。 2、可信主机认证。 3、Kerberos认证。 在后文中将会讨论这些方式的优缺点。 sftp 它是在SSH上的一个独立文件传输工具，操作与ftp类似，可以在一个会话中调用多个命令 进行文件拷贝和处理，而scp每次调用时都要打开一个会话。sftp不区分ascii and binary传 输模式，只使用二进制模式，因此，如果使用它在windows and unix之间拷贝ascii文件， 那么就不能正确转换行结束符。 第三章 SSH内幕 SSH协议的主要特性和优点： 1、 使用强加密技术来保证数据的私密性。端到端通信用随机密钥进行加密，随机密钥为会 话进行安全协商，会话结束后被丢弃。支持的算法有ARCFOUR,Blowfish,DES,IDEA,3DES等。 2、通信完整性，确保通信不会被修改。SSH-2基于MD5 AND SHA-1的加密hash算法。 3、认证，即发送者和接收者的身份证明。客户机和服务器双向认证。 4、授权，即对账号进行访问控制。 5、使用转发或隧道技术对其它基于tcp/ip的会话进行加密。支持三种转发，tcp端口转发， X转发，代理转发。 SSH可以防止的攻击 1、网络窃听，SSH通信是加密的，即使截获会话内容，也不能将其解密。 2、名字服务和IP伪装，SSH通过加密验证服务器主机身份可避免这类风险。 3、连接劫持，SSH的完整性检测负责确定会话在传输过程是否被修改，如果被修改过，就 关闭连接。 4、 中间人攻击，SSH利用两种方法防止这种攻击，第一种是服务器主机认证。除非攻击者 已经成功攻击了服务器主机，获得服务器的私有主机密钥。第二种是限制使用容易受到 这种攻击的认证方法，密码认证容易受到中间人攻击，而公钥和基于主机的/RhostsRSA 则对中间人攻击可以免疫。 5、插入攻击，这种攻击可以客户和服务器之间发送的正文数据流之间插入任意数据。ssh1  1.2.25后和openssh的所有版本都专门进行了设计，来检测并防止这种攻击。这种检测程序 增大了插入攻击的难度，但是并不能完全防止。ssh2使用强加密完整性检测手段来防止这个 问题。可以用3DES算法来防止这种攻击。 SSH不能防止的攻击 1、 密码崩溃，密码认证是一种脆弱的认证形式，尽量使用公钥认证方式。如果必须要密码 认证，可考虑使用S/Key之类的一次性密码机制。 2、 IP AND TCP攻击，由于SSH是在TCP之上进行操作的，因此容易受到针对TCP和IP 缺陷而发起的攻击。SYN flood,TCP不同步和TCP劫持等。只能通过更低层的防护措施 来保护。 3、流量分析。 4、隐秘通道。 5、粗心大意。安全是一个过程，而不是一个产品，不要认为装上SSH就安全了。 第四章 SSH的安装和编译时配置 第五章 服务器范围的配置 sshd可以在三个层次上进行配置，第一层次是上面的安装和编译时配置；第二层次是本章的 服务器范围配置；第三层次是每账号配置（第八章），前者是服务器编译时就指定了包含哪 些特定功能，不包含哪些功能，后者则是由终端用户来修改自已账号所使用的服务器的行为。 以普通用户身份运行SSH服务器 1、得到管理员许可。 2、生成一个主机密钥。 $ ssh-keygen -N '' -b 1024 -f ~/myserver/hostkey 生成hostkey and hostkey.pub两个文件 3、选择端口号。 要选择大于或等1024的端口，因为只有超级用户才有权使用小于 1024的端口 。 4、创建服务器配置文件(可选) 可创建自已的配置文件，否则，服务就使用内建的特性或使 用服务器范围的配置文件。 启动服务器方式：$ sshd -h ~/myserver/hostkey -p 2345 -f ~/myserver/config 由普通用户运行的服务器有一些缺点： 1、由于不是由root的uid下运行，因此只能连接到用户自已的账号上。 2、它需手工调用，不能自启动。 3、 日志用户没权看到，因为服务器把它写到syslog日志系统中，但可用调试模式运行服务 4、 器，这样信息就显示在终端机上。 服务器配置文件 SSH1 AND OPENSSH的配置文件通常是/etc/sshd\_config,而SSH2的配置文件通常是 /etc/ssh2/sshd2\_config。 以非缺省的配置文件来调用sshd，可以使用命令 -f 选项。$ sshd -f /usr/local/ssh/my\_config。 以\*号作为标号的语名都会被sshd忽略。 修改完配置后，修改不会马上影响到服务器，可重启服务器或发送SIGHUP信号。$ kill -HUP  `cat /etc/sshd.pid` 如果在命令行改动了配置，用SIGHUP信号重启服务器不能覆盖命令行的配置，它的优先 级高。 主机密钥，sshd主机密钥向SSH客户端唯一地标识SSH服务器。主机密钥保存在一对文件 中，一个文件包含私钥，一个文件包含公钥。对于SSH1 AND OPENSSH，它是 /etc/ssh\_host\_key。文件位置可用HostKey关键字修改 HostKey /usr/local/ssh/key。服务器的 公钥保存在.pub文件中。 OPENSSH服务器有一个SSH-2的主机密钥，缺省是在/etc/ssh\_host\_dsa\_key。其位置可用 HostDsaKey /usr/local/openssh/key2修改。 对于SSH2，如果服务器是由超级用户运行，缺省私钥文件是/etc/ssh2/hostkey。如果是其它 用户运行，则在~/.ssh2/hostkey。可用HostKeyFile /etc/local/ssh/key修改。公钥是hostkey.pub 文件，可用publicHostKeyFile关键字修改。 可以用命令行选项指定私钥文件：$ sshd -h /usr/local/ssh/key。 随机数种子文件 如果用户的系统中有一个随机位源，如/dev/urandom，那么OPENSSH就不能创建随机数种 子文件。 服务器配置文件 通常是/etc/sshd\_config /etc/ssh2/ssh2\_config，可以用-f 选项重新指定其配置文件。可用于一 台机器运行多个sshd的情况。 每账号认证文件。~/.ssh/authorized\_keys 。  文件权限 由于配置文件的权限设置错误，会降低系统的安全性，所以可以设置StrictModes YES/NO 关键字，对sshd的重要文件和目录的权限进行检查，如果检查失败，服务器就拒绝对该用 户的SSH连接。 在多宿主主机上，使用ListenAddress xxx.xxx.xxx.xxx来限制SSH只监听一个网络接口。 空闲连接时间，IdleTimeout xx 如果是0代表什么都不做，保持连接。否则，超时就断开连 接，可以用s,m,h,d,w表示。 KeepAlive YES/NO 可以断开失效的连接，比如客户端崩溃。 失败的登录 LogingGraceTime 60 限定用户必须在60秒内成功进行认证。 0值表禁用。命令行选项用  -g PasswordGuesses 5 如果连接请求使用密码认证，那么sshd2就只允许客户端尝试五次。 （SSH2) 限制并发连接 MaxConnections 32 最大32个连接（SSH2） 逆向IP映射 SSH2服务器可以根据客户端地址进行逆向DNS查询，以确保客户端的地址就是这个地址。 如果检查失败，连接拒绝。 RequireReverseMapping yes (ssh2) 转发 AllowTcp-Forwarding yes  X11Forwarding yes 服务器密钥生成 该密钥用来对客户端/服务器之间的通信进行保护。是临时的，永远不会保存在磁盘上。服 务器在启动时生成这个密钥。并以固定的周期重新生成。缺省长度是768位，最小为512， 可以ServerKeyBits 2048 指定长度。用KeyRegenerationInterval 1200指定周期。 命令行选 项：-k 加密算法 Ciphers any 允许所有支持算法。其它算法包括 3des-cbc,blowfish-cbc,twofish-cbc,arcfour,none mac算法 MAC关键字可以让用户选择sshd2进行完整性检测所使用的算法，称为消息认证代码。用 于sshd2的有hmac-sha1,hmac-md5,hmac-md5-96。 在OPENSSH中，用户可以用Protocol关键字选择支持SSH-1 AND SSH-2,1 代表SSH-1,2 代 表SSH-2, 1，2 代表都支持。 允许用户登录：认证和访问控制 1、认证负责对发起连接请求的用户的身份进行验证。 密码认证 PasswordAuthentication yes AllowedAuthentications password (ssh2) 公钥认证 RSAAuthentication yes (ssh1,openssh/1) AllowedAuthentications publickey (ssh2) DEAAuthentication yes (openssh/2) Rhosts认证，可信主机认证通过检查远程主机名和相关用户名来实现对客户端的身份认证。 RhostsAuthentication yes (ssh1,openssh) IgnoreRhosts yes (ssh1,ssh2,openssh)不使用系统的hosts.equiv and ~/.rhosts，使用SSH专用 的/etc/shosts.equiv and ~/.shosts 强可信主机认证。 RhostsRSAAuthentication yes ssh1,openssh AllowedAuthentications hostbased ssh2 提取已知名主机的公钥 UserKnownHosts no ssh2 IgnoreUserKnownHosts yes openssh PGP认证 Kerberos认证 S/key认证 SecurID认证 PAM认证 2、访问控制负责允许或禁止来自特定用户、机器在或INTERNET域的SSH连接到服务器 上。 通常，只要设置正确，任何账号都可以接收SSH连接，这种访问权限可以使用服务器关键 字AllowUsers and DenyUsers覆盖。 AllowUsers smith 如果配置文件中单独出现一个AllowUsers,后面没有任何内容，就表示禁止所有没提到的用 户连接，如果没有AllowUsers这个关键字，则所有用户都可连接。DenyUsers表示禁止连接。 可用通配符， 可以用组访问控制AllowGroups DenyGroups 主机名访问控制 AllowHosts hostname DenyHosts hostname 超级用户的访问控制 sshd对超级用户专门使用一种特殊的访问机制，PermiRootLogin来允许或禁止使用SSH来 访问root账号。 显示用户的欢迎信息 PrintMotd yes/no default is yes 显示邮件信息 CheckMail yes/no default is yes 空密码 PermitEmptyPasswords yes/no 如果使用密码认证，并且有个账号没有设定密码，那么服务 器就可以拒绝访问这个账号 如果系统中存在/etc/nologin，那么sshd就只允许root用户登录，其它都不能登录。因此，touch  /etc/nologin是把访问权只授给系统管理员的一种快速方法。这样并不需要重新配置ssh。也 不需要重启ssh。 子系统 定义和调用远程命令的一个抽象层，用户可以通过在客户端命令行中给出命令来调用远程命 令，如：ssh server.examply.com /bin/tar c /home 这个命令调用tar,远程把/home拷贝到磁带上。 子系统是服务器机器上预定义的一组远程命令，这样就可以方便地执行。在服务器配置文档 中定义：subsystem backups /bin/tar c /home，要在服务器上运行该命令，可以使用 -s 选项。 ssh -s backups server.example.com。缺省情况下，sshd\_config中定义了一个子系统,subsystem  sftp 。不要删除这个子系统，这是scp2和sftp必须的。 日志 Fascisl Logging mode 把调试信息打印到系统日志文件中，可以使用FascistLogging关键字 启用。 调试模式 可以使用-d命令选项启用 安静模式 只能输出严重错误，而不能输出 普通日志，可以使用QuietMode关键字或-q 命令行启用。 openssh中的日志配置关键字是SyslogFacility and LogLevel。SyslogFacility设置syslog的 facility（KERN,DAEMON,USER,AUTH,MAIL等），LogLevel记录日志提供的详细程度，该 值从低到高顺序是：QUIET,FATAL,ERROR,INFO,VERBOSE,DEBUG,使用DEBUG会侵犯用 户的隐私权，这个级别只能用于诊断，而不能用于普通操作。 第六章 密钥管理与代理 一个身份标识由两部份组成，分别称为私钥（Private Key）和公钥（Public Key），合称一个 密钥对。 SSH1,SSH2 AND OPENSSH身份标识文件的格式各不相同。 SSH1缺省设置中，私钥存储在文件identity中，公钥存储在文件identity.pub中。这个密钥 对存放在~/.ssh目录下，在使用前，把公钥拷贝到服务器上的一个认证文件里，如SSH1 AND  OPENSSH中的~/.ssh/authorized\_keys，此后，当客户请求连接到你的服务器上的帐号时，就 会用一个私钥作为证明身份的标识，服务器则在authorized\_keys文件中寻找与之匹配的公 钥。 SSH2密钥对文件的命名通常是根据该密钥使用的加密算法的性质来起的，例如一个用DSA 加密的1024位密钥生成时其缺省文件名是id\_dsa\_1024\_a and id\_dsa\_1024\_a.pub。用户必须 把私钥放在identification文件中，缺省是~/.ssh2/目录，一个私钥在这个文件中占一行，在公 钥认证中，每一行的开头都有一个关键字IdKey，后跟一个私钥文件名。如： IdKey id\_dsa\_1024\_a IdKey my-other-ssh2-key 在服务器端的认证文件~/.ssh2/authorization中，SSH2不包含公钥的实际拷贝，只是把公钥 文件列出来，前面用关键字kEY标识。如： Key id\_dsa\_1024\_a.pub Key something-else.pub 这样维护起来更容易，更方便。 OPENSSH对SSH-1连接使用的标识和授权文件与SSH1完全相同，对于SSH-2连接，缺省 的密钥则存储在~/.ssh/id\_dsa(private key)和~/.ssh/id\_dsa.pub(public key)中，服务器上的授权 文件是~/.ssh/authorized\_keys。 创建身份标识用ssh-keygen命令。 OPENSSH的ssh-keygen可以支持SSH1中相应程序的所有功能和选项，它还新增了为SSH-2 连接生成DSA密钥的能力。 $ ssh-keygen -t dsa -b 1024 -e -x 选项可把OPENSSH格式的密钥存储格式转换成SSH2格式 (直接回车，按提示输入转 换文件名) -i -X 选项把SSH2格式的密钥转换成OPENSSH格式。这样可用SSH2客户端连接 OPENSSH服务器。(直接回车，按提示输入转换文件名) -y 如果不小心删掉了OPENSSH的公钥，可用-y恢复。 -l 可计算出公钥的指纹，指纹是鉴别位置不同的两个密钥是否相同的常用的一种密码学技 术。这项技术用于不可能逐位对比两个密钥的情况。OPENSSH AND SSH2都能计算指纹。 它是根据密钥计算出来一个长度较短的值，它是原理与校验和类似，用于验证一串信息（在 我们的例子中是密钥）的不可替换性。 启动代理 1、单shell方式，使用当前登录的shell. $ eval `ssh-agent` 2、子shell方式，派生出一个子shell，并继承父shell的某些环境变量。 $ ssh-agent $SHELL 不要想当然运行 $ ssh-agent，这样客户端是无法有代理联系的。删除代理用kill命令就可以 了。 SSH-1 与 SSH-2代理的兼容性 SSH-1的代理不能处理SSH-2的代理，SSH-2的代理却可以处理SSH-1的代理请求。 第七章 客户端的高级用法 $ ssh -a -p 220 -c blowfish -l sally -i myself server.example.com 可通过配置文件简化命令 输入 Host myserver ForwardAgent no Port 220 Cipher blowfish User sally IdentifyFile myself HostName server.example.com 配置了该文件后，就可以在客户端简单地输入 $ ssh myserver 全局文件和本地文件 全局文件通常是由管理员创建的，用来维护整台计算机上的客户行为。该文件通常位于 /etc/ssh\_config or /etc/ssh2/ssh2\_config。每个客户也可以在自已账号中创建本地客户配置文 件，通常是~/.ssh/config or ~/.ssh2/ssh2\_config。本地配置文件优先级高于全局配置文件。命 令行选项的优先级又高于本地配置文件。 scp有关命令参数 -r 递归拷贝目录 -p 保持文件权限和时间戳 -u 拷贝完成后删除源文件 -d 防止覆盖文件 -n 描述操作 -q 不显示统计信息 第八章 每账号服务器配置 这种配置可以让SSH服务器区分每个服务器上的各个用户。利用服务器目标账号的认证文 件（authorized\_keys）配置。局限有两点，它不能覆盖编译时配置和服务器范围配置所采用 的安全措施，第二，如果使用公钥认证，每账号配置是十分灵活的，如果采用可信主机和密 码认证，提供的选择范围就很小。 基于公钥的配置 1、认证文件的完全格式 依次包含三项内容，一些选项，公钥和注释。多个选项用逗号分 开。 2、用于限制客户端可以在服务器上调用的程序的强制命令。如客户端请求执行/bin/ls命令， 而服务器端强制命令却运行/bin/who命令，它十分有用，可用于加强安全性，了为授权提供 方便。如要让你的助手每次连接上来都运行email命令程序pine，则可以这样： command= "/usr/local/bin/pine" ...key...。最多可给每个密钥关联一个强制命令，要关联多个命 令可把这些命令放入远程主机的一个脚本中，并将该脚本作为强制命令运行。如果强制命令 有shell出口，那会存在安全问题，等于没有强制，客户可以运行任何程序。以下规则可以 用来判断一个程序是否适合用作强制命令。 a、 避免使用具有shell出口的程序（如：文件编辑器vi,Emacs; 分页程序 more,less;调用分 b、 页的man,新闻阅读程序rn，邮件阅读程序pine以及调试程序adb，非交互的程序find,xargs c、 等）。 b、避免使用编译器，解释器或其它可以让用户生成并运行任意执行代码的程序。 c、可创建或删除文件的程序，如cp,mv,rm,scp,ftp等。 d、避免使用setuid or setgid的程序，特别是setuid 是root的程序。 d、 如果使用脚本作为强制命令，就要遵循编写安全脚本的传统规则，在一个脚本之内，要 限制使用相对路径作为搜索路径，应该使用绝对路径来调用所有的程序；不要盲目地把 用户提供的字符串作为命令来执行；不要让脚本执行任何setuid的工作。不要调用具有 shell出口的程序。 e、考虑使用受限shell。 f. 为一个单独的、专用的SSH密钥（不要用你登录的那个密钥）关联一个强制命令。这样 不会影响你的登录能力就可以方便地禁用该密钥。 g、禁用不必要的SSH特性。如no-port-forwarding,no-agent-forwarding,no-pty(禁用tty分配)。 几个常用的强制命令介绍： 使用定制消息拒绝连接：command="/bin/echo Sorry,buddy,but you've terminated!" ...key... 千万 不要用more and less之类的分页程序 显示命令菜单：利用脚本 检查客户端的原始程序：command="/bin/echo you tried to invoke  $SSH\_ORIGINAL\_COMMAND" ...key... $SSH\_ORIGINAL\_COMMAND环境变量保存着 客户连接时运行的原始命令。 限制客户端的原始命令，创建一个脚本，该脚本根据$SSH\_ORIGINAL\_COMMAND内容选 择不同的操作。 把客户端的原始命令记录在日志中：也是根据$SSH\_ORIGINAL\_COMMAND变量来做的一 个脚本。脚本内容如一： #!/bin/sh if [ -n "$SSH\_ORIGINAL\_COMMAND" ] then echo "`/bin/date`: $SSH\_ORIGINAL\_COMMAND" >> $HOME/ssh-command-log exec $SSH\_ORIGINAL\_COMMAND fi 3、限制来自特定主机的连接。由from选项完成。from="client.example.com" ...key...。强制 连接必须来自client.example.com。否则断开。from="!client.example.com" ...key...表示拒绝该 地址的连接。支持通配符。SSH2不支持该功能。但可以通过$SSH2\_CLIENT变量提取客户 ip，编写强制命令脚本达到同样效果。 4、为远程程序设置环境变量。environment="EDITOR=emacs" ...key...,为每个连接修改默认的 环境变量。 5、设置空闲超时时间，这样如果客户端用户不再发送数据就强制将其断开。idle-timeout=5m, 该选项会覆盖服务器范围配置。 6、对到达的SSH连接禁用某些特性，例如端口转发和tty分配。禁用tty分配会使客户端没 有交互会话的能力。$SSH\_TTY变量可显示tty情况。 用户的rc文件 ssh服务器会在每个连接到达时调用shell脚本/etc/sshrc，用户可把一些登录后想运行的脚本 写到这个文件中。用户可以在自已的目录下定义类似的脚本，如果有自定义的~/.ssh/rc则 /etc/sshrc则不会执行。 第九章 端口转发与X转发 使用SSH对其它应用程序在别的TCP端口上建立的TCP/IP传输进行加密和解密，这一过 程称为端口转发，其绝大多数操作都是透明的，功能非常强大。TELNET,SMTP,NNTP,IMAP 和一些基于TCP的不安全协议都可变得安全，只要将其连接通过SSH转发即可。端口转发 有时又叫做隧道传输。 例子：一个运行IMAP的服务器S，在家里的主机H上运行一个email程序，现在想用ssh 保护IMAP连接。 $ ssh -L2001:localhost:143 S -L 表明是本地转发，，上面这命令可完成登录到S的功能，现在的这个SSH会话同时将H 的2001端口转发到S的143端口，在退出会话之前一直有效。并告知email程序使用被转 发的端口，通常邮件程序连接服务器的143端口，即套接字 （S，143）。现在要令其连接 本地主机H自已的2001端口，也就是套接字（localhost,2001）。创建本地转发时可以不用-L 选项，而在客户端配置文件中用LocalForward关键字。 LocalForward 2001 localhost:143 ssh1 openssh LocalForward "2001:localhost:143" ssh2 -g 选项和GatewayPorts yes/no关键字可以配置除本地主机外的机器是否能使用本地的端口 转发功能。出于安全问题，该禁止这个功能。 远程转发端口与本地转发几乎完全相同，只是方向相反。此时服务器在本地，转发连接由远 程主机发起。假设你已登录进服务器S。则 $ ssh -R2001:localhost:143 H -R代表远程转发。命令执行后，一条从远程主机H的2001端口到服务器S的143端口的安 全隧道就建立起来了。H上的任何程序都可以通过连接（localhost,2001）来使用此安全隧道。 而在客户端配置文件中用RemoteForward关键字. RemoteForward 2001 S:143 ssh openssh RemoteForward "2001:S:143" ssh2 $ ssh -o ClearAllForwardings=yes server.example.com 阻止第二条命令建立隧道  本地转发与远程转发 在本地转发中，应用程序客户端与监听端同SSH客户端在一起，应用程序服务器与连接端 同SSH服务器在一起。 在远程转发中，应用程序客户端与监听端同SSH服务器在一起，应用程序服务器与连接端 同SSH客户端在一起。 无远程登录的端口转发，连接时带-f参数。一次性转发用-fo，ssh1与openssh不支持这个功 能，但可用以下语句实现： $ ssh -f -L2001:localhost:143 server sleep 10 终止 如果一个SSH会话现在还在活动当中，终止会出错。在SSH2中，如果退出有活动转发连 接的会话，该会话会依然打开，但会转到后台运行。直到转发连接终止。ssh1 and openssh 与此相反，如果要断开还在活动的会话，会出现警告，仍然在前台运行。 TIME\_WAIT问题。在某些情况下，TCP连接断连时，其一端的套接字在很短一段时间内变 得不可用，所以在断连过程结束前不能把该端口用于TCP转发。   第十章 推荐配置 服务器范围配置 1、禁用其它访问方式。 关掉r-命令，方法如下：删除/etc/hosts.equiv文件，或改为只读空文件。禁用rshd,rlogind and  rexecd，通过修改inetd.conf文件。 2、/etc/sshd\_config配置 HostKey /etc/ssh\_host\_key PidFile /etc/sshd.pid RandomSeed /etc/ssh\_random\_seed StrictModes yes 要求用户保护其与SSH有关的文件及目录 Umask 0077 保证所有SSHD1创建是文件和目录都只能由其所有者（SSHD 运行的 UID）读取 port 22  ListenAddress 0.0.0.0 IdleTimeout 15m 15分钟空闲超时 KeepAlive yes 客户机死机时连接将会终止，而不会长期挂起。 LoginGraceTime 30 登录时成功认证的时限为30秒。 ServerKeyBits 768 服务器密钥长度 KeyRegenerationInterval 3600 服务器密钥一小时重新生成一次 PasswordAuthentication no 禁用密码认证 RhostsAuthentication no 禁用可信主机认证 RhostsRSAAuthentication no 禁用增强可信主机认证 RSAAuthentication yes 启用公钥认证 IgnoreRhosts yes 完全禁止SSHD使用.rhosts文件 IgnoreRootRhosts yes  UseLogin no 禁用UseLogin，防止万一使用了其它登录程序 AllowHosts xxx 根据需要设置 SilentDeny yes 任何被DenyHosts拒绝的连接都不会向用户返回消息，不必告诉攻击 者发生了什么，可以增加了排错的难度。 PermitRootLogin nopwd 允许超级用户通过SSH连接，但不能用密码认证 FascistLogging no 禁用FascistLogging日志方式，因为它会在日志中记录用户特定信息，对 攻击者有用。 QuietMode no 禁用QuietMode日志方式，使用日志更详细，敏感度更低 AllowTcpForwarding yes 允许tcp端口转发和X转发，保护其它的tcp连接 X11Forwarding yes 3、/etc/ssh2/sshd2\_config配置 HostKeyFile /etc/ssh2/hostkey PublicHostKeyFile /etc/ssh2/hostkey.pub RandmoSeedFile /etc/ssh2/random\_seed UserConfigDirectory IdentityFile AuthorizstionFile StrictModes yes port 22 listenAddress 0.0.0.0 KeepAlive yes RequireReverseMapping no LoginGraceTime 30 由于sshd2没有设置服务器密钥的位数的关键字，用户得在启动时使用 -b 选项 $ sshd2 -b  1024 .... AllowedAuthentications publickey RequiredAuthentications publickey IgnoreRhosts yes UserKnownHosts no 禁用该项可防止用户给未知主机提供信任权限。 PermitRootLogin nopwd Ciphers any 不能选none QuietMode no VerboseMode yes Ssh1Compatibility no 禁用SSH-1兼容模式 #Ssh1Path /usr/local/ssh1/sshd1 出于实用原因，也可以启用此模式，之前要指明ssh1服 务器可执行文件位置 4、每账号配置 对于SSH1 AND OPENSSH，~/.ssh/authorized\_keys中的每一个密钥都必须用适当的选项进 行限制，from选项限制只能从特定的主机访问特定的密钥。例如，假设文件中包含你家那 台pc（myhome.isp.net）的公钥,而其它机器根本不可能用那个密钥来认证，我们就可以明确 限定这一关系：from = "myhome.isp.net" ...key...。还要对适当的密钥设置空闲超时时间：from  = "myhome.isp.net" ，idle-timeout=5m ...key...。 最后，考虑每一个密钥是否需要对到达的连接使用端口转发，代理转发以及分配tty等，如 果不需要，就可以分别用no-port-forwarding,no-agent-forwarding and no-pty禁用这些特性。 5、密钥管理 至少创建1024位长的用户密钥，并用好的口令对密钥进行保护。 6、客户端配置 离开正在运行的ssh客户端时，一定要用密码保护。 在客户端配置文件中启用某些安全功能，并将其设置为最强 Host \* FallBackToRsh no 禁止使用不安全的r-命令（ssh2）没有这个问题。 UseRsh no GatewayPorts no 禁止远程客户端连接本地的转发端口 StrictHostKeyChecking ask 在主机密钥发生变化时提醒你。请求你的处理意见。 配置“/etc/ssh/ssh\_config”文件　 “/etc/ssh/ssh\_config”文件是OpenSSH系统范围的配置文件，允许你通过设置不同的选项 来改变客户端程序的运行方式。这个文件的每一行包含“关键词－值”的匹配，其中“关键 词”是忽略大小写的。下面列出来的是最重要的关键词，用man命令查看帮助页（ssh　(1)） 可以得到详细的列表。　 编辑“ssh\_config”文件（vi　/etc/ssh/ssh\_config），添加或改变下面的参数：　 #　Site-wide　defaults　for　various　options　 Host　\*　 ForwardAgent　no　 ForwardX11　no　 RhostsAuthentication　no　 RhostsRSAAuthentication　no　 RSAAuthentication　yes　 PasswordAuthentication　yes　 FallBackToRsh　no　 UseRsh　no　 BatchMode　no　 CheckHostIP　yes　 StrictHostKeyChecking　no　 IdentityFile　~/.ssh/identity　 Port　22　 Cipher　blowfish　 EscapeChar　~　 下面逐行说明上面的选项设置：　 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 “$HOME/.ssh/known\_hosts”文件，并且一旦计算机的密匙发生了变化，就拒绝连接。　 IdentityFile　~/.ssh/identity　 “IdentityFile”设置从哪个文件读取用户的RSA安全验证标识。　 Port　22　 “Port”设置连接到远程主机的端口。　 Cipher　blowfish　 “Cipher”设置加密用的密码。　 EscapeChar　~　 “EscapeChar”设置escape字符。　 配置“/etc/ssh/sshd\_config”文件　 “/etc/ssh/sshd\_config”是OpenSSH的配置文件，允许设置选项改变这个daemon的运行。 这个文件的每一行包含“关键词－值”的匹配，其中“关键词”是忽略大小写的。下面列出 来的是最重要的关键词，用man命令查看帮助页（sshd　(8)）可以得到详细的列表。　 编辑“sshd\_config”文件（vi　/etc/ssh/sshd\_config），加入或改变下面的参数：　 #　This　is　ssh　server　systemwide　configuration　file.　 Port　22　 ListenAddress　192.168.1.1　 HostKey　/etc/ssh/ssh\_host\_key　 ServerKeyBits　1024　 LoginGraceTime　600　 KeyRegenerationInterval　3600　 PermitRootLogin　no　 IgnoreRhosts　yes　 IgnoreUserKnownHosts　yes　 StrictModes　yes　 X11Forwarding　no　 PrintMotd　yes　 SyslogFacility　AUTH　 LogLevel　INFO　 RhostsAuthentication　no　 RhostsRSAAuthentication　no　 RSAAuthentication　yes　 PasswordAuthentication　yes　 PermitEmptyPasswords　no　 AllowUsers　admin　 下面逐行说明上面的选项设置：　 Port　22　 “Port”设置sshd监听的端口号。　 ListenAddress　192.168.1.1　 “ListenAddress”设置sshd服务器绑定的IP地址。　 HostKey　/etc/ssh/ssh\_host\_key　 “HostKey”设置包含计算机私人密匙的文件。　 ServerKeyBits　1024　 “ServerKeyBits”定义服务器密匙的位数。　 LoginGraceTime　600　 “LoginGraceTime”设置如果用户不能成功登录，在切断连接之前服务器需要等待的时间（以 秒为单位）。　 KeyRegenerationInterval　3600　 “KeyRegenerationInterval”设置在多少秒之后自动重新生成服务器的密匙（如果使用密匙）。 重新生成密匙是为了防止用盗用的密匙解密被截获的信息。　 PermitRootLogin　no　 “PermitRootLogin”设置root能不能用ssh登录。这个选项一定不要设成“yes”。　 IgnoreRhosts　yes　 “IgnoreRhosts”设置验证的时候是否使用“rhosts”和“shosts”文件。　 IgnoreUserKnownHosts　yes　 “IgnoreUserKnownHosts”设置ssh　daemon是否在进行RhostsRSAAuthentication安全验证 的时候忽略用户的“$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 [email=user@host]user@host[/email] 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。 -------------------------------------------------------------------------------- |