网络管理 linux端口映射Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/644/2021_2022__E7_BD_91_E7_BB_9C_E7_AE_A1_E7_c103_644794.htm ssh -C -g root@127.0.0.1 -L 5000:61.235.139.123:5000 输入机器的root密码 后台执行：ssh -C -f -N -g root@127.0.0.1 -L 5000:61.235.139.123:5000 另： ssh -C -f -N -g -R remote_port:local:port user@remotehost 可 以将远端服务器一个端口remote_port绑定到本地端口port，其中-C是进行数据压缩，-f是后台操作，只有当提示用户名密码的时候才转向前台。-N是不执行远端命令，在只是端口转发时这条命令很有用处。-g 是允许远端主机连接本地转发端口。-R表明是将远端主机端口映射到本地端口。如果是-L，则是将本地端口映射到远端主机端口。 关于ssh端口转发的深入实例 2007-05-13 17:02 Thursday, 5. April 2007, 13:44:15 转自geminis@http://floss.zoomquiet.org/data/20070104103806/ ssh的三个强大的端口转发命令： ssh -C -f -N -g -L listen_port:DST_Host:DST_port user@Tunnel_Host ssh -C -f -N -g -R listen_port:DST_Host:DST_port user@Tunnel_Host ssh -C -f -N -g -D listen_port user@Tunnel_Host -f Fork into background after authentication. 后台认证用户/密码，通常和-N连用，不用登录到远程主机。 -p port Connect to this port. Server must be on the same port. 被登录的ssd服务器的sshd服务端口。 -L port:host:hostport 将 本地机(客户机)的某个端口转发到远端指定机器的指定端口. 工作原理是这样的, 本地机器上分配了一个 socket 侦听 port 端口, 一旦这个端口上有了连接, 该连接就

经过安全通道转发出去, 同时远程主机和 host 的 hostport 端口建立连接. 可以在配置文件中指定端口的转发. 只有 root 才能转发特权端口. IPv6 地址用另一种格式说明: port/host/hostport -R port:host:hostport 将 远程主机(服务器)的某个端口转发到本地端指定机器的指定端口. 工作原理是这样的, 远程主机上分配了一个 socket 侦听 port 端口, 一旦这个端口上有了连接, 该连接就经过安全通道转向出去, 同时本地主机和 host 的 hostport 端口建立连接. 可以在配置文件中指定端口的转发. 只有用 root 登录远程主机才能转发特权端口. IPv6 地址用另一种格式说明: port/host/hostport -D port 指定一个本地机器“ 动态的 ’ 应用程序端口转发. 工作原理是这样的, 本地机器上分配了一个 socket 侦听 port 端口, 一旦这个端口上有了连接, 该连接就经过安全通道转发出去, 根据应用程序的协议可以判断出远程主机将和哪里连接. 目前支持 SOCKS4 协议, 将充当 SOCKS4 服务器. 只有 root 才能转发特权端口. 可以在配置文件中指定动态端口的转发. -C Enable compression. 压缩数据传输。 -N Do not execute a shell or command. 不执行脚本或命令，通常与-f连用。 -g Allow remote hosts to connect to forwarded ports. 在-L/-R/-D参数中，允许远程主机连接到建立的转发的端口，如果不加这个参数，只允许本地主机建立连接。注：这个参数我在实践中似乎始终不起作用，参见III) iptables实现端口转发的过程 设我们有一台计算机,有两块网卡,eth0连外网,ip为1.2.3.4.eth1连内网,ip为192.168.0.1.现在需要把发往地址1.2.3.4的81端口的ip包转发到ip地址192.168.0.2的8180端口,设置如下: 1. iptables -t nat -A PREROUTING -d 1.2.3.4 -p tcp -m tcp --dport 81 -j DNAT --to-destination192.168.0.2:8180 2. iptables -t

nat -A POSTROUTING -s 192.168.0.0/255.255.0.0 -d 192.168.0.2 -p tcp -m tcp --dport 8180 -j SNAT --to-source 192.168.0.1 真实的传输过程如下所示: 假设某客户机的ip地址为6.7.8.9,它使用本机的1080端口连接1.2.3.4的81端口,发出的ip包源地址为6.7.8.9,源端口为1080,目的地址为1.2.3.4,目的端口为81. 主 机1.2.3.4接收到这个包后,根据nat表的第一条规则,将该ip包的目的地址更该为192.168.0.2,目的端口更该为8180,同时在连接跟 踪表中创建一个条目,(可从/proc/net/ip_conntrack文件中看到),然后发送到路由模块,通过查路由表,确定该ip包应发送到 eth1接口.在向eth1接口发送该ip包之前,根据nat表的第二条规则,如果该ip包来自同一子网,则将该ip包的源地址更该为 192.168.0.1,同时更新该连接跟踪表中的相应条目,然后送到eth1接口发出. 此时连接跟踪表中有一项: 连接进入: src=6.7.8.9 dst=1.2.3.4 sport=1080 dport=81 连接返回: src=192.168.0.2 dst=6.7.8.9 sport=8180 dport=1080 是否使用: use=1 而从192.168.0.2发回的ip包,源端口为8180,目的地址为6.7.8.9,目的端口为1080,主机1.2.3.4的TCP/IP栈接收到该 ip包后,由核心查找连接跟踪表中的连接返回栏目中是否有同样源和目的地址和端口的匹配项,找到后,根据条目中的记录将ip包的源地址由 192.168.0.2更该为1.2.3.4, 源端口由8180更该为81,保持目的端口号1080不变. 这样服务器的返回包就可以正确的返回发起连接的客户机,通讯就这样开始. 