 在网络管理中，对于DNS服务的管理是一项基础性的工作。随着用户规模的扩大，频繁地手工修改DNS的区域数据库文件不是一件轻松的工作。关于动态DNS(DDNS)的研究逐渐引起了人们的关注，不同的平台都推出了自己的解决方案。本文将详细介绍Linux环境下DDNS的解决方案，即由Internet Software Consortium（ISC）开发的BIND-DNS和DHCP(Dynamic Host Configure Protocol，动态主机配置协议)协同工作，进而共同实现DDNS的方法。 在Linux下实现动态DNS不仅需要Bind 8以上的DNS软件，还要有DHCP Server v3.0以上版本，因为只有3.0以上的版本才完全实现了对DDNS的支持。因此，本文的实现环境采用Slackware Linux 9.0作为DDNS服务器，其上同时运行DNS和DHCP服务，其中DNS Server采用Bind 9.2.2，DHCP Server采用DHCP Server v3.0pl2。 下面详细介绍Linux环境下安全、动态DNS的实现方法。 创建密钥 要实现DNS的动态更新，首先要考虑的是怎样保证安全地实现DDNS。由ISC给出的方法是创建进行动态更新的密钥，在进行更新时通过该密钥加以验证。为了实现这一功能，需要以root身份运行以下命令： root@slack9:/etc# dnssec-keygen -a HMAC-MD5 -b 128 -n USER myddns Kmyddns. 157 37662 上述dnssec-keygen命令的功能就是生成更新密钥，其中参数-a HMAC-MD5是指密钥的生成算法采用HMAC-MD5；参数-b 128是指密钥的位数为128位；参数-n USER myddns是指密钥的用户为myddns。 该命令生成的

一对密钥文件如下： -rw 1 root root 48 Jan 14 18:26 Kmyddns. 157 37662.key -rw 1 root root 81 Jan 14 18:26 Kmyddns. 157 37662.private 可以查看刚生成的密钥文件内容： root@slack9:/etc# cat Kmyddns. 157 37662.key myddns.INKEY02157 4gEF1Mkmn5hrlwYUeGJV3g== root@slack9:/etc# cat Kmyddns. 157 37662.private Private-key-format: v1.2 Algorithm: 157 (HMAC_MD5) Key: 4gEF1Mkmn5hrlwYUeGJV3g== 仔细阅读该密钥文件就会发现，这两个文件中包含的密钥是一样的，该密钥就是DHCP对DNS进行安全动态更新时的凭据。后面需要将该密钥分别添加到DNS和DHCP的配置文件中。 修改DNS的主配置文件 密钥生成后就要开始对/etc/named.conf文件进行编辑修改，主要目的是将密钥信息添加到DNS的主配置文件中。本文给出修改后的/etc/named.conf的一个实例：options { directory "/var/named". file://指定区域数据库文件的存放目录 }. zone "." IN { type hint. file "caching-example/named.ca". }. zone "localhost" IN { type master. file "caching-example/localhost.zone". allow-0update { none. }. }. zone "0.0.127.in-addr.arpa" IN { type master. file "caching-example/named.local". allow-0update { none. }. }. key myddns { algorithm HMAC-MD5.SIG-ALG.REG.INT. file://指明生成密钥的算法 secret 4gEF1Mkmn5hrlwYUeGJV3g==. file://指明密钥 }. zone "tcbuu.cn" IN { type master. file "tcbuu.cn". file://正向区域文件名tcbuu.cn，后文会用到该文件 allow-0update { key myddns. }. file://指明采用key myddns作为密钥的用户可以动态更新该区域“ tcbuu.cn ” }. zone "1.22.10.in-addr.arpa" IN { type master. file "tcbuu.cn.arpa".//反向

区域文件名tcbuu.cn allow-0update { key myddns. }. file://指明采用key myddns作为密钥的用户可以动态更新该区域“ 1.22.10.in-addr.arpa ”}. 在/etc/named.conf中可以定义多个区域，只要在允许动态更新的区域中增加allow-0update { key myddns. }指令，即可实现动态更新，并且只有拥有key myddns实体（在本文的实现中该实体就是拥有同样密钥的DHCP服务器）才能实现对该区域进行安全地动态更新。相比原来只限定IP地址的方法，该方法要安全得多。 至此完成对DNS服务器的配置，可以执行#named运行DNS服务。