**DDNS配置实例【DHCP+DNS=DDNS】**

**DDNS配置实例（DHCP+DNS=DDNS）**

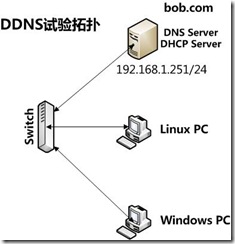
在网络管理中，维护DNS服务器是一项很基本的工作。但是，随着网络规模的不断扩大，频繁的去修改DNS区域数据文件，那也将会是一件很麻烦的事情。因此，动态DNS就应运而生。

动态DNS（DDNS）需要DNS和DHCP来协同工作。Linux下也可以实现DDNS，不过DNS需要Bind8以上的版本，DHCP需要3.0以上的版本。

本文用RedHat9来实现DDNS，Bind版本是bind-9.2.1-16，DHCP版本是dhcp-3.0pl1-23。DNS和DHCP使用一台服务器。

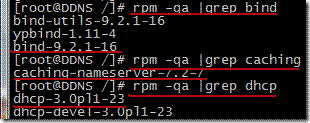
后记讲使用Cent OS 5.1实现DDNS。

**1. DDNS试验拓扑**

[](http://img1.51cto.com/attachment/200811/7/363003_1226036086iO0P.jpg)

**2. 安装DNS和DHCP软件包**

挂载系统盘，进入RPM包目录，使用**rpm –ivh**来安装bind包和dhcp包。其中caching-nameserver包是用来安装name.ca、name.local等区域文件的，如果不安装此包，那么就要手写或者从别的地方down这些文件了。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036087nvl9.gif)

**3. 配置DNS服务**

**① 编辑DNS服务主配置文件named.conf，默认在/etc/目录下。**

**vi /etc/named.conf**

**options {**

**directory "/var/named";**

**};**

**zone "." IN {**

**type hint;**

**file "named.ca";**

**};**

**zone "localhost" IN {**

**type master;**

**file "localhost.zone";**

**allow-update { none; };**

**};**

**zone "0.0.127.in-addr.arpa" IN {**

**type master;**

**file "named.local";**

**allow-update { none; };**

**};**

**zone "bob.com" IN {**

**type master;**

**file "bob.com.dns";**

**allow-update { none; };**

**};**

**zone "1.168.192.in-addr.arpa" IN {**

**type master;**

**file "bob.com.rev";**

**allow-update { none; };**

**};**

**include "/etc/rndc.key"；**

**② 建立区域数据文件**

因为安装了caching-nameserver这个包，所以像named.ca、named.local和localhost.zone这三个区域数据文件都有了。现在来创建bob.com域的正向和反向区域数据文件。

**vi /var/named/bob.com.dns**

**$TTL 86400**

**$ORIGIN bob.com.**

**@    IN   SOA   bob.com.   root.bob.com. (**

**20081103 ;     serial**

**120 ;          refresh**

**14400 ;      retry**

**3600000 ;  expiry**

**86400 ) ;    minimum**

**IN   NS     bob.com.**

**dns                IN    A      192.168.1.251**

**www             IN    A      192.168.1.250**

**vi /var/named/bob.com.rev**

**$TTL 86400**

**@    IN    SOA   bob.com.   root.bob.com. (**

**20081103 ;     Serial**

**120 ;          Refresh**

**14400 ;      Retry**

**3600000 ;  Expire**

**86400 ) ;    Minimum**

**IN   NS      bob.com.**

**251               IN  PTR     dns.bob.com.**

**250               IN  PTR** [**[url]www.bob.com[/url]**](http://www.bob.com/)**.**

**chown named.named /var/named/bob.com.\***

**③ 测试DNS服务**

使用**service named start**命令启动DNS服务。

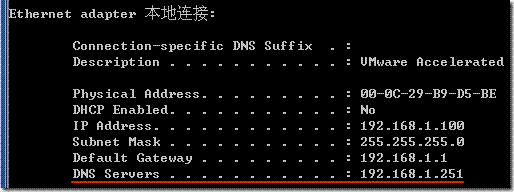
[named](http://img1.51cto.com/attachment/200811/7/363003_1226036087xR9b.gif)

在防火墙中将UDP和TCP的53端口开放。

然后使用**service iptables restart**命令重启防火墙。

Windows PC测试：

在Windows PC上将DNS指向192.168.1.251。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036088Ldq8.gif)

在Windows PC上使用**nslookup**命令来解析DNS服务器中的域名。

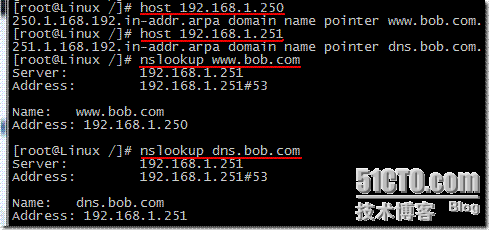
[](http://img1.51cto.com/attachment/200811/7/363003_1226036092vMn9.gif)

Linux  PC测试：

在Linux PC上将DNS指向192.168.1.251。

[resolv](http://img1.51cto.com/attachment/200811/7/363003_1226036093dpgJ.gif)

在Linux PC上使用**host**命令和**nslookup**命令来解析DNS服务器中的域名。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036095Z59Z.gif)

**4. 配置DHCP服务**

**① 编辑DHCP服务主配置文件dhcpd.conf，默认在/etc/目录下。**

**vi /etc/dhcpd.conf**

**ddns-update-style interim;**

**ignore client-updates;**

**default-lease-time 604800;**

**max-lease-time 864000;**

**option domain-name “bob.com”;**

**option domain-name-servers 192.168.1.251;**

**option time-offset -18000;**

**subnet 192.168.1.0 netmask 255.255.255.0 {**

**range 192.168.1.100 192.168.1.200;**

**option broadcast-address 192.168.1.255;**

**option routers 192.168.1.1;**

**}**

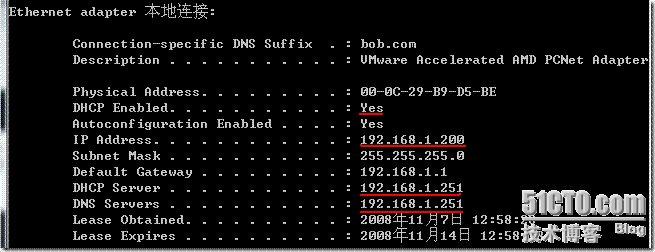
**② 测试DHCP+DNS服务**

使用**service dhcpd start**命令启动DHCP服务。

[dhcpd](http://img1.51cto.com/attachment/200811/7/363003_1226036099Ifnx.gif)

Windows PC测试：

在Windows PC上设置动态获取IP和DNS。

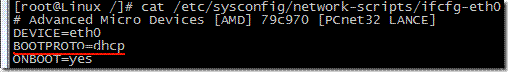
[](http://img1.51cto.com/attachment/200811/7/363003_1226036099beOK.gif)

在Windows PC上使用**nslookup**命令来解析DNS服务器中的域名。

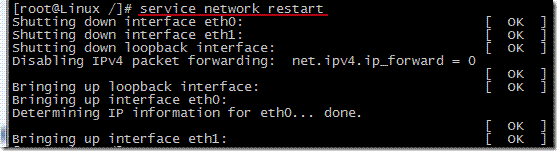
[](http://img1.51cto.com/attachment/200811/7/363003_1226036102746y.gif)

Linux PC测试：

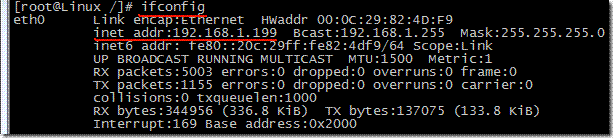
在Linux PC上使用动态获取IP和DNS。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036104Tov6.gif)

使用**service network restart**命令重启网卡。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036107oRl8.gif)

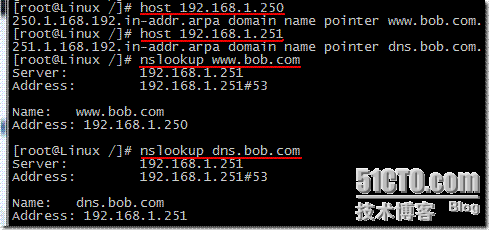
使用**ifconfig**命令查看Linux PC获取到的IP地址。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036108EOTS.gif)

使用**cat /etc/resolv.conf**查看Linux PC获取到的DNS地址。

[dhcp](http://img1.51cto.com/attachment/200811/7/363003_1226036110Conn.gif)

在Linux PC上使用**host**命令和**nslookup**命令来解析DNS服务器中的域名。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036112tRQZ.gif)

**5. 配置安全的DDNS**

**① 创建密钥**

在DDNS服务器中以root身份进行：

使用**dnssec–keygen –a HMAC-MD5 –b 128 –n USER bobddns**命令来生成密钥。

**dnssec-keygen：**用来生成更新密钥。

**-a HMAC-MD5：**采用HMAC-MD5加密算法。

**-b 128：**生成的密钥长度为128位。

**-n USER bobddns：**密钥的用户名为bobddns。

[dnssec](http://img1.51cto.com/attachment/200811/7/363003_1226036114MI7U.gif)

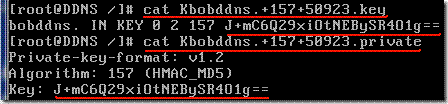
密钥生成后，会在当前目录下自动生成两个密钥文件Kbobddns.+157+xxx.key和Kbobddns.+157+xxx.private。

[ls两个密钥文件](http://img1.51cto.com/attachment/200811/7/363003_1226036114ebPs.gif)

查看两个密钥文件的内容：

**cat Kbobddns.+157+50923.key**

**cat Kbobddns.+157+50923.private**

[](http://img1.51cto.com/attachment/200811/7/363003_1226036115oOQo.gif)

两个密钥文件中的128位密钥是一致的。需记住这一串密钥字符串，后面将会用到。

**② 添加密钥信息到DNS主配置文件中**

**vi /etc/named.conf**

添加：

**key bobddns {**

**algorithm hmac-md5;**

**secret J+mC6Q29xiOtNEBySR4O1g==;**

**};**

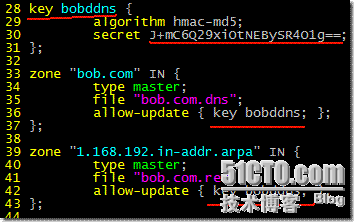
algorithm：指明生成密钥的算法。

secret：指明密钥串。

将bob.com区域中的allow-update { none; }中的“none”改成“**key bobddns**”;

将1.168.192.in-addr.arpa区域中的allow-update { none; }中的“none”也改成“**key bobddns**”。

将“none”改成“key bobddns”的意思是指明采用“key bobddns”作为密钥的用户可以动态更新“bob.com”区域。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036116s7RE.gif)

**③ 添加密钥信息到DHCP主配置文件中**

**vi /etc/dhcpd.conf**

添加：

**key bobddns {**

**algorithm hmac-md5;**

**secret J+mC6Q29xiOtNEBySR4O1g==;**

**}**

**zone bob.com. {**

**primary 192.168.1.251;**

**key bobddns;**

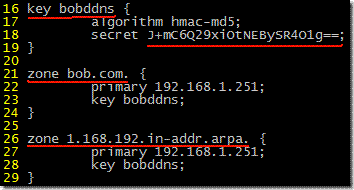
**}**

**zone 1.168.192.in-addr.arpa. {**

**primary 192.168.1.251;**

**key bobddns;**

**}**

**[](http://img1.51cto.com/attachment/200811/7/363003_12260361173WVv.gif)**

**注意：域名后面的“.”千万不能少了，还有在每个“}”后面都没有“;”，这两点和named是不一样的，请注意。**

**④ 重启DHCP和DNS服务**

**service dhcpd restart**

**service named restart**

[](http://img1.51cto.com/attachment/200811/7/363003_1226036118FtYA.gif)

**⑤ 测试DDNS**

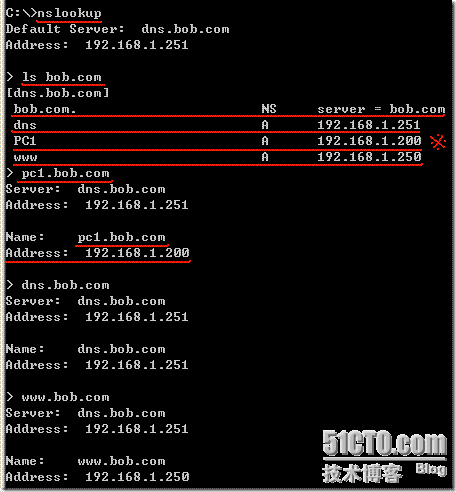
Windows PC测试：

使用**hostname**命令查看主机名。

[windowspc](http://img1.51cto.com/attachment/200811/7/363003_1226036119SjBH.gif)

使用**ipconfig/release**和**ipconfig/renew**命令重新向DHCP服务器获取IP。

使用**nslookup**命令测试。使用**ls bob.com**命令，发现PC1已经被添加到了bob.com这个域中了。然后使用**pc1.bob.com**能解析到自己的IP地址了。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036119jtSE.gif)

Linux PC测试：

使用hostname命令查看主机名。

[Linuxpc](http://img1.51cto.com/attachment/200811/7/363003_1226036120SZDt.gif)

在Linux PC上的/etc目录下新建一个DHCP客户端文件，文件名为dhclient.conf。

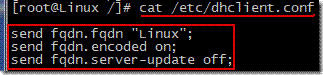
**vi /etc/dhclient.conf**

加入：

**send fqdn.fqdn "Linux";**

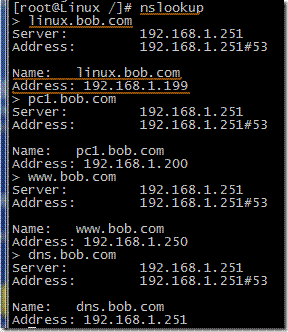
**send fqdn.encoded on;**

**send fqdn.server-update off;**

[](http://img1.51cto.com/attachment/200811/7/363003_1226036121WnoX.gif)

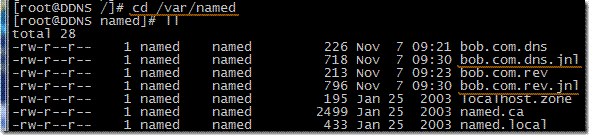
使用**dhclient**命令立即启动DHCP客户端。

使用**nslookup**命令测试。使用linux.bob.com能解析到自己的IP地址了，使用pc1.bob.com也能解析到Windows PC的IP地址了。

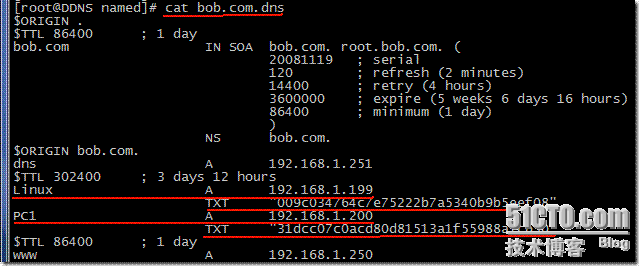
[](http://img1.51cto.com/attachment/200811/7/363003_1226036122tRmK.gif)

DDNS服务器测试：

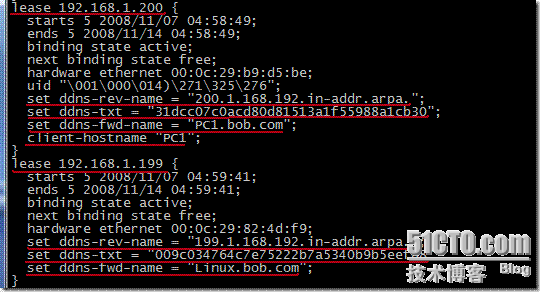
在重启完DHCP和DNS服务后，如果正常，会在/var/named/目录下生成两个**.jnl**二进制格式区域文件。这两个文件是当前正在工作的区域文件的运行时文件，所有动态更新的记录首先会反应到这两个文件中，然后经过大约15分钟的时间才会将更新内容添加到区域数据文件中。在本文的例子中，区域数据文件就是“bob.com.dns”和“bob.com.rev”。

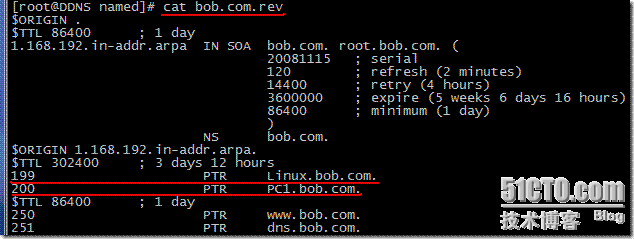
[](http://img1.51cto.com/attachment/200811/7/363003_1226036125V1zg.gif)

查看更新后的区域数据文件，系统会将手写的区域数据文件格式更改成它自己的格式。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036126Kth2.gif)

说明：以上区域文件的书写格式与更新前相比变化较大，说明该文件已经被更新过了。Windows PC和Linux PC的主机记录已经在区域数据文件中自动添加进来了。这里还要说明的是，在动态更新的客户端PC1和Linux的A纪录下多了一条同名的TXT类型的纪录。TXT类型纪录是BIND-DNS和DHCP专门用来实现DDNS的辅助性资源纪录，它的值是哈希标示符字符串，该字符串的值还可以在DHCP租约文件**/var/lib/dhcp/dhcpd.leases**中找到。

[](http://img1.51cto.com/attachment/200811/7/363003_1226036131hItw.gif)

[](http://img1.51cto.com/attachment/200811/7/363003_12260361344Kdk.gif)

**后记：Cent OS 5.1实验DDNS成功。**

刚开始我使用Cent OS 5.1做这个实验，但是没有成功。named和dhcpd这两个服务都能够正常的启动，但是两个区域数据文件的.jnl文件死活不出现，这样就没有达到动态更新的目的。因为以前我使用Red Hat 9将这个实验做了出来，所以，现在我就先使用Red Hat 9来做这个实验，成功了。

接下来，我又重新使用Cent OS 5.1来做这个DDNS实验，但是结果还是一样。我以为是chroot搞的鬼，就将bind-chroot-9.3.3-10.el5这个包卸载了，但是结果还是不成功。那问题就不在chroot上了，我又重装上了bind-chroot-9.3.3-10.el5。

后来，我比对了一下Red Hat 9和Cent OS 5的区域数据文件的属主（Red Hat 9 的区域数据文件默认在/var/named/目录下；Cent OS 5的区域数据文件默认在/var/named/chroot/var/named/目录下。），发现Red Hat 9的named目录的属主和属组都是named，而Cent OS 5的named目录的属主是root，属组是named，并且named目录下的区域数据文件的属主和属组与named目录一样。因此，我将named目录的属主和属组都改成了named后，并且重启named服务，使用updatedb命令重刷系统数据库后，在/var/named/chroot/var/named/目录下两个.jnl文件出现了。此时使用Windows PC和Linux PC测试都成功了。http://yuanbin.blog.51cto.com/editor/icons/etc_14.gif真的很高兴，做了几遍终于搞定了。原来是文件及目录的权限搞怪，使用ps aux可以看到named服务的管理者是named。

如果使用Cent OS或者Red Hat 企业版做这个实验，其他步骤和上面一样，只要将name目录的属主改成named即可搞定。

chown -R named.named /var/named/chroot/var/named/

祝愿大家能够做成功！