第11章 Linux网络基本配置

网卡配置（文件名）、常见命令、systemctl

人民邮电出版社
POSTS & TELECOM PRESS
精品系列

要完成网络配置工作，可以修改相应的配置文件、使用网络命令以及通过图形界面进行设置。要管理网络服务，可以使用服务配置工具、ntsysv命令以及chkconfig和service命令来启动和停止服务。为了保证Linux系统的安全，本章最后介绍了一些提高Linux系统安全性能和增强Linux系统安全保护的方法。

11.1 Linux网络配置文件

用户可以在Fedora Core 17系统中编辑相应的网络配置文件来完成网络配置工作，下面详细介绍这些网络配置文件。

11.1.1 /etc/sysconfig/network-scripts/ifcfg-eth0文件

在Fedora Core 17系统中，系统网络设备的配置文件保存在/etc/sysconfig/network-scripts目录下，其中文件ifcfg-eth0包含第一块网卡的配置信息，文件ifcfg-eth1包含第二块网卡的配置信息，文件ifcfg-lo包含回路IP地址信息。

11.1.2 /etc/resolv.conf文件

文件/etc/resolv.conf是由域名解析器（resolver，一个根据主机名解析IP地址的库）使用的配置文件。

11.1.3 /etc/host.conf文件

文件/etc/host.conf指定如何解析主机名，Fedora Core 17系统通过解析器库来获得主机名对应的IP地址。

11.1.4 /etc/sysconfig/network文件

文件/etc/sysconfig/network用来指定服务器上的网络配置信息。

11.1.5 /etc/hosts文件

当计算机启动时，在可以查询DNS以前，计算机需要查询一些主机名到IP地址的匹配。这些匹配信息存放在/etc/hosts文件中。在没有域名服务器的情况下，系统上的所有网络程序都通过查询该文件来解析对应于某个主机名的IP地址。

11.1.6 /etc/services文件

文件/etc/services定义了Linux系统中所有服务的名称、协议类型、服务的端口等信息。

11.2 Linux网络命令

在Linux系统中提供了大量的网络命令用于网络配置、网络测试以及网络诊断，如ifconfig，ping，netstat，traceroute，arp以及tcpdump等。

11.2.1 traceroute

使用traceroute命令可以显示数据包到目标主机之间的路径。

命令语法：

traceroute [-dFlnrvx][-f<存活数值>][-g<网关>...][-i<网络界面>][-m<存活数值>][-p<通信端口>][-s<来源地址>][-t<服务类型>][-w<超时秒数>][主机名称或IP地址][数据包大小]

【例11.1】跟踪从本地计算机到www.163.com网站的路径。

```
[root@PC-LINUX ～]# traceroute www.163.com
traceroute to www.163.com (220.181.28.50), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  1.036 ms  0.922 ms  0.829 ms
 2  58.41.132.1 (58.41.132.1)  2.054 ms  1.990 ms  2.195 ms
 3  222.72.255.153 (222.72.255.153)  1.698 ms  1.664 ms  1.658 ms
 4  218.1.4.29 (218.1.4.29)  1.933 ms  1.877 ms  1.971 ms
 5  218.1.0.198 (218.1.0.198)  2.672 ms  2.398 ms  2.503 ms
 6  202.101.63.194 (202.101.63.194)  2.472 ms  2.467 ms  2.400 ms
 7  (202.97.34.61)  23.957 ms  23.979 ms  23.876 ms
 8  (218.30.25.45)  24.117 ms  24.373 ms  24.134 ms
 9  (218.30.25.238)  24.764 ms  24.601 ms  24.557 ms
10  (220.181.16.138)  29.313 ms  37.606 ms  38.297 ms
11  (220.181.17.58)  27.850 ms  35.903 ms  36.844 ms
12  (220.181.28.50)  24.424 ms  20.385 ms  57.837 ms
```

11.2.2 ifconfig

使用ifconfig命令可以显示或设置计算机网卡的IP地址。

命令语法：

ifconfig [网络设备][down up -allmulti -arp -promisc][add<地址>][del<地址>][<硬件地址>][mtu<字节>][netmask<子网掩码>][IP地址]

【例11.2】 配置网卡eth0的IP地址， 同时激活该设备。

```
[root@PC-LINUX ~]# ifconfig eth0 192.168.0.100 netmask
255.255.255.0 up
```

【例11.3】 配置网卡eth0别名设备eth0:1的IP地址。

```
[root@PC-LINUX ~]# ifconfig eth0:1 192.168.0.3
```

【例11.4】 激活网卡eth0:1设备。

```
[root@PC-LINUX ~]# ifconfig eth0:1 up
```

【例11.5】 查看网卡eth0网络接口的配置。

```
[root@PC-LINUX ~]# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.100  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe75:d61e  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:75:d6:1e  txqueuelen 1000  (Ethernet)
        RX packets 5711  bytes 489075 (477.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2625  bytes 312104 (304.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000
//查看该网卡IP地址是192.168.0.100,MAC地址是00:0c:29:75:d6:1e
```

【例11.6】 查看所有的网卡网络接口配置。

```
[root@PC-LINUX ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.100  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::20c:29ff:fe75:d61e  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:75:d6:1e  txqueuelen 1000  (Ethernet)
        RX packets 5660  bytes 484973 (473.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2596  bytes 307846 (300.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 19  base 0x2000

eth0:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.3  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 00:0c:29:75:d6:1e  txqueuelen 1000  (Ethernet)
        device interrupt 19  base 0x2000
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 16436
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 0  (Local Loopback)
        RX packets 94  bytes 8664 (8.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 94  bytes 8664 (8.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 2a:e1:6e:43:1b:99  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

【例11.7】 禁用网卡eth0:1设备。

```
[root@PC-LINUX ~]# ifconfig eth0:1 down
```

11.2.3 ping

使用ping命令可用于测试与目标计算机之间的连通性。

命令语法：

ping [-dfnqrRv][-c<完成次数>][-i<间隔秒数>][-I<网络界面>][-l<前置载入>][-p<范本样式>][-s<数据包大小>][-t<存活数值>][主机名称或IP地址]

【例11.8】 测试与网站www.ak.com的连通性。

```
[root@PC-LINUX ～]# ping www.ak.com
PING www.ak.com(191.161.1.28) 56(84) bytes of data.
64 bytes from 191.161.1.28: icmp_req=0 ttl=64 time=1.49 ms
64 bytes from 191.161.1.28: icmp_req=1 ttl=64 time=0.873 ms
64 bytes from 191.161.1.28: icmp_req=2 ttl=64 time=1.00 ms
64 bytes from 191.161.1.28: icmp_req=3 ttl=64 time=0.440 ms
……
//在Linux系统中用该命令会不间断地返回ICMP数据包，要停止测试按[Ctrl+c]键
```

【例11.9】测试与192.168.0.222计算机的连通性，每次发送的ICMP数据包大小为128字节。

```
[root@PC-LINUX ~]# ping -s 128 192.168.0.222
PING 192.168.0.111 (192.168.0.222) 128(156) bytes of data.
136 bytes from 192.168.0.222: icmp_req=1 ttl=64 time=1.38 ms
136 bytes from 192.168.0.222: icmp_req=2 ttl=64 time=0.645 ms
136 bytes from 192.168.0.222: icmp_req=3 ttl=64 time=0.426 ms
136 bytes from 192.168.0.222: icmp_req=4 ttl=64 time=0.358 ms
136 bytes from 192.168.0.222: icmp_req=5 ttl=64 time=0.401 ms
……
```

【例11.10】测试与192.168.0.5计算机的连通性，要求返回4个ICMP数据包。

```
[root@PC-LINUX ~]# ping -c 4 192.168.0.5
PING 192.168.0.111 (192.168.0.5) 56(84) bytes of data.
64 bytes from 192.168.0.5: icmp_req=1 ttl=64 time=0.527 ms
64 bytes from 192.168.0.5: icmp_req=2 ttl=64 time=0.352 ms
64 bytes from 192.168.0.5: icmp_req=3 ttl=64 time=0.429 ms
64 bytes from 192.168.0.5: icmp_req=4 ttl=64 time=0.340 ms

--- 192.168.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3038ms
rtt min/avg/max/mdev = 0.340/0.412/0.527/0.074 ms
```

11.2.4 netstat

使用netstat命令可用于显示网络状态的信息。

命令语法：

netstat [-acCeFghilMnNoprstuvVwx][-A<网络类型>][--ip]

【例11.11】 显示网络接口状态信息。

```
[root@PC-LINUX ~]# netstat -i
Kernel Interface table
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP
TX-OVR Flg
eth0      1500     5923      0      0 0          2752      0      0      0 BMRU
lo       16436       97      0      0 0            97      0      0      0 LRU
virbr0    1500        0      0      0 0             0      0      0      0 BMU
```

【例11.12】 显示内核路由表信息。

```
[root@PC-LINUX ~]# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         192.168.0.1     0.0.0.0         UG        0 0          0 eth0
192.168.0.0     *               255.255.255.0   U         0 0          0 eth0
192.168.122.0   *               255.255.255.0   U         0 0          0 virbr0
```

【例11.13】 显示UDP传输协议的连接状态。

```
[root@PC-LINUX ～]# netstat -u
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address               Foreign Address
State
udp        0      0 192.168.0.222:filenet-pa    192.168.0.100:domain
ESTABLISHED
udp        0      0 192.168.0.222:filenet-cm    192.168.0.100:domain
ESTABLISHED
udp        0      0 192.168.0.222:filenet-re    192.168.0.100:domain
ESTABLISHED
udp        0      0 192.168.0.222:32805         192.168.0.100:domain
ESTABLISHED
```

11.2.5 arp

使用arp命令可用于增加、删除和显示arp缓存。

命令语法：

arp -a

【例11.14】 查看系统arp缓存。

```
[root@PC-LINUX ~]# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.5              ether   00:50:56:c0:00:01   C
```

【例11.15】 添加一个IP地址和MAC地址的对应记录。

```
[root@PC-LINUX ~]# arp -s 192.168.0.99 00:60:08:27:CE:B2
[root@PC-LINUX ~]# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.0.5              ether   00:50:56:c0:00:01   C                     eth0
192.168.0.99             ether   00:60:08:27:ce:b2   CM
//可以看到刚才添加的静态ARP记录
```

【例11.16】 删除一个IP地址和MAC地址的对应缓存记录。

```
[root@PC-LINUX ～]# arp -d 192.168.0.99
[root@PC-LINUX ～]# arp
Address          Hwtype          HWaddress                Flags Mask            Iface
192.168.0.5      ether           00:50:56:C0:00:01              C               eth0
```

11.2.6 tcpdump

使用tcpdump命令可以监视TCP/IP连接，并直接读取数据链路层的数据包头，可以指定哪些数据包被监视以及哪些控制要显示格式。

命令语法：

Tcpdump [-adeflnNOpqStvx][-c<数据包数目>][-dd][-ddd][-F<表达文件>][-i<网络界面>][-r<数据包文件>][-s<数据包大小>][-tt][-T<数据包类型>][-vv][-w<数据包文件>][输出数据栏位]

【例11.17】使用指定的网络接口eth0读取数据链路层的数据包头。

```
[root@PC-LINUX ～]# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:40:01.324515 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq
1401759532                                          :1401759648, ack 1000464709, win
190, length 116
10:40:01.325282 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 116:168, a
ck 1, win 190, length 52
10:40:01.325621 IP 192.168.0.5.49585 > PC-LINUX.ssh: Flags [.], ack 168, win 16
370, length 0
10:40:01.325856 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 168:284, a
ck 1, win 190, length 116
10:40:01.326319 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 284:336, a
ck 1, win 190, length 52
```

10:40:01.326626 IP 192.168.0.5.49585 > PC-LINUX.ssh: Flags [.], ack 336, win 16328, length 0
10:40:01.328341 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 336:516, ack 1, win 190, length 180
10:40:01.328810 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 516:568, ack 1, win 190, length 52
10:40:01.329217 IP 192.168.0.5.49585 > PC-LINUX.ssh: Flags [.], ack 568, win 16270, length 0
10:40:01.329407 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 568:716, ack 1, win 190, length 148
10:40:01.330041 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 716:768, ack 1, win 190, length 52
10:40:01.330351 IP 192.168.0.5.49585 > PC-LINUX.ssh: Flags [.], ack 768, win 16220, length 0
10:40:01.330520 IP PC-LINUX.ssh > 192.168.0.5.49585: Flags [P.], seq 768:900, ack 1, win 190, length 132
……

11.3 图形界面配置网络

在Fedora Core 17系统中使用“网络配置”工具可以用来配置计算机的IP地址、子网掩码、网关地址、DNS服务器的IP地址等信息。在图形化桌面环境中，单击面板上的“活动”→“应用程序”→“其他”→“网络连接”，弹出如图11-1所示界面。

图11-1　“网络连接”界面


正在编辑 系统 eth0
连接名称(I)
系统 eth0
自动连接(A)
有线
802.1x 安全性
IPv4 设置
IPv6 设置
方法(M)：
手动
地址
地址
子网掩码
网关
添加(A)
192.168.0.100
24
192.168.0.1
删除(D)
DNS 服务器：
202.96.209.5
搜索域(E)：
DHCP 客户端 ID：
需要 IPv4 地址完成这个连接
路由(R)...
对所有用户可用
取消(C)
保存(S)...

图11-2 设置IPv4

11.4 管理网络服务

管理Linux系统服务的方法有很多。可以根据不同的服务、系统配置以及对Linux系统的掌握程度来决定使用哪一种管理方法。可以使用以下3种不同的方法启动或停止这些服务。

（1）ntsysv：基于文本的程序。它允许为每个运行级别配置引导时要启动的服务。对于独立服务而言，改变不会立即生效。

（2）systemctl命令：Fedora 17中新的管理服务的命令，用来替换chkconfig和service命令。

（3）chkconfig和service命令：允许在不同运行级别启动和关闭服务的命令行工具。

11.4.1 ntsysv命令

使用ntsysv命令可以配置让服务在系统启动时自动启动或停止。

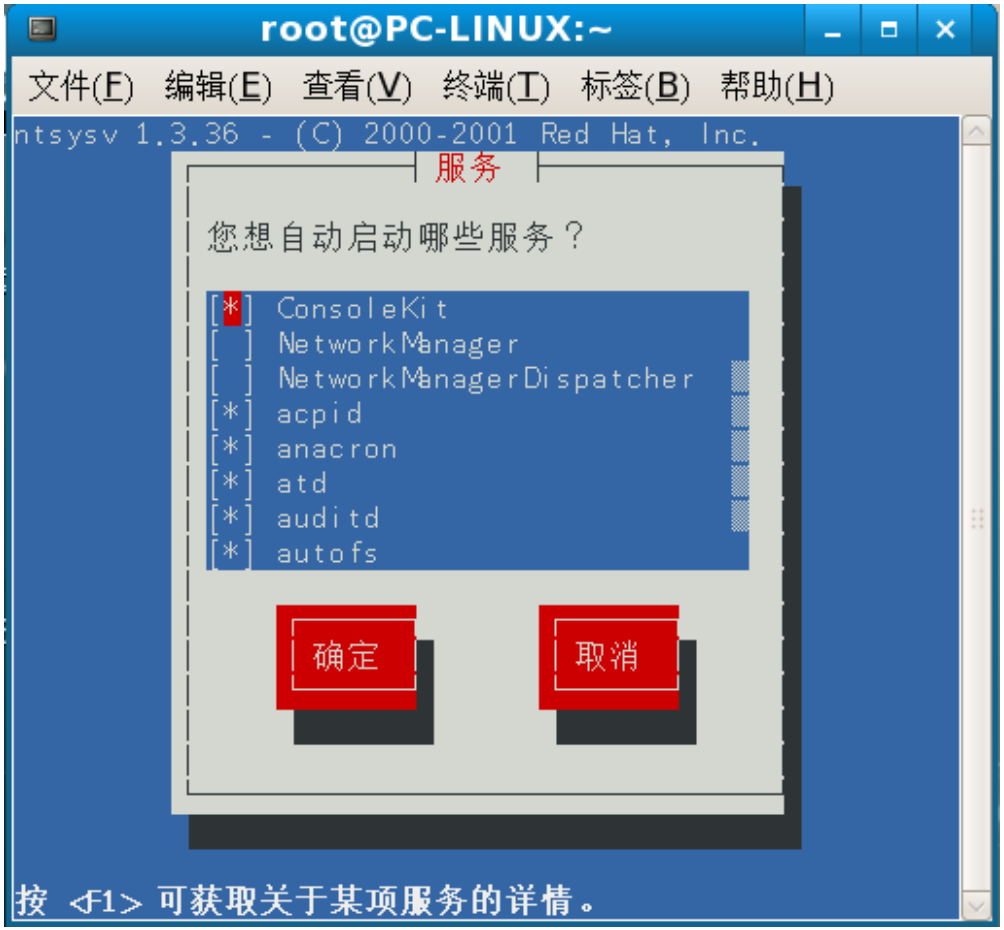

图11-3　ntsysv运行界面

11.4.2 systemctl命令

在Fedora 17系统中使用的是新的systemd系统和服务管理程序。systemctl是系统服务管理命令，它实际上将service和chkconfig这两个命令组合到一起使用。

命令语法：
systemctl 选项 [服务名].service
命令中各选项的含义如下。
start：表示启动服务。
stop：表示停止服务。
status：表示查看服务状态。
restart：表示重新启动服务。
reload：表示加载服务配置文件。
enable：表示开机自动启动服务。
disable：表示开机禁止启动服务。
is-enabled：表示查看服务是否开机自动启动。
list-units --type=service：显示所有已启动的服务。

【例11.18】 启动named服务。

```
[root@PC-LINUX ~]# systemctl start named.service
```

【例11.19】查看named服务当前状态。

```
[root@PC-LINUX ~]# systemctl status named.service
named.service - Berkeley Internet Name Domain (DNS)
          Loaded: loaded (/usr/lib/systemd/system/named.service; disabled)
          Active: active (running) since Thu, 30 May 2013 09:31:38 +0800;
1min 1                                                    1s ago
        Process: 3073 ExecStop=/bin/sh -c /usr/sbin/rndc stop > /dev/null
2>&1                                                    || /bin/kill -TERM $MAINPID
(code=exited, status=0/SUCCESS)
        Process: 3083 ExecStart=/usr/sbin/named -u named $OPTIONS
(code=exited,                                                 status=0/SUCCESS)
        Process: 3081 ExecStartPre=/usr/sbin/named-checkconf -z
/etc/named.conf                                                 (code=exited,
status=0/SUCCESS)
       Main PID: 3084 (named)
          CGroup: name=systemd:/system/named.service
                  └ 3084 /usr/sbin/named -u named
```

```
May 30 09:31:38 PC-LINUX named[3084]: open: /etc/rndc.key: file not found
May 30 09:31:38 PC-LINUX named[3084]: couldn't add command
channel ::1#953:...nd
May 30 09:31:38 PC-LINUX named[3084]: managed-keys-zone: loaded serial 3
May 30 09:31:38 PC-LINUX named[3084]: zone 0.in-addr.arpa/IN: loaded serial 0
May 30 09:31:38 PC-LINUX named[3084]: zone 1.0.0.127.in-addr.arpa/IN: loade...
0
May 30 09:31:38 PC-LINUX named[3084]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0....
0
May 30 09:31:38 PC-LINUX named[3084]: zone localhost.localdomain/IN:
loaded... 0
May 30 09:31:38 PC-LINUX named[3084]: zone localhost/IN: loaded serial 0
May 30 09:31:38 PC-LINUX named[3084]: all zones loaded
May 30 09:31:38 PC-LINUX named[3084]: running
```

【例11.20】停止named服务。
[root@PC-LINUX ~]# systemctl stop named.service
【例11.21】重新启动named服务。
[root@PC-LINUX ~]# systemctl restart named.service
【例11.22】重新加载named服务配置文件。
[root@PC-LINUX isolinux]# systemctl reload named.service
【例11.23设置named服务开机自动启动。
[root@PC-LINUX ~]# systemctl enable named.service
ln –s '/usr/lib/systemd/system/named.service' '/etc/systemd/system/multi-user.target.wants/named.service'
【例11.24】查询named服务是否开机自动启动。
[root@PC-LINUX isolinux]# systemctl is-enabled named.service
enabled
【例11.25】停止named服务开机自动启动。
[root@PC-LINUX ~]# systemctl disable named.service
[root@PC-LINUX ~]# systemctl is-enabled named.service
disabled

【例11.26】查看所有已启动的服务。

```
[root@PC-LINUX ~]# systemctl list-units --type=service
UNIT                    LOAD   ACTIVE SUB     JOB DESCRIPTION
abrt-ccpp.service       loaded active exited      Install ABRT coredump hook
abrt-oops.service       loaded active running     ABRT kernel log watcher
abrt-vmcore.service     loaded active exited      Harvest vmcores for ABRT
abrtd.service           loaded active running     ABRT Automated Bug Reportin
accounts-daemon.service loaded active running     Accounts Service
acpid.service           loaded active running     ACPI Event Daemon
atd.service             loaded active running     Job spooling tools
auditd.service          loaded active running     Security Auditing Service
avahi-daemon.service    loaded active running     Avahi mDNS/DNS-SD Stack
bluetooth.service       loaded active running     Bluetooth Manager
colord-sane.service     loaded active running     Daemon for monitoring attac
colord.service          loaded active running     Daemon for managing, instal
console-kit-daemon.service loaded active running  Console Manager
console-...m-start.service loaded active exited   Console System Startup Logg
```

11.4.3 chkconfig和service命令

在Linux系统下可以使用chkconfig和service命令控制服务的启动、停止和重新启动，两者之间的区别在于使用service命令控制服务可以马上生效，而使用chkconfig命令控制服务需要等计算机重新启动后才会生效，在Fedora 17系统中这2个命令的功能有所限制，建议使用systemctl命令。

1. chkconfig命令

使用chkconfig命令主要用来设置下次重启计算机以后启动、停止服务，使用chkconfig命令不会立即自动启动或停止一项服务。

命令语法：

chkconfig --list [服务名]

【例11.27】 查看各种不同的运行等级中SysV服务的状况。

```
[root@PC-LINUX ~]# chkconfig --list
```

注意：该输出结果只显示 SysV 服务，并不包含原生 systemd 服务。SysV 配置数据可能被原生 systemd 配置覆盖。

```
ceph            0:关    1:关    2:关    3:关    4:关    5:关    6:关
dc_client       0:关    1:关    2:关    3:关    4:关    5:关    6:关
dc_server       0:关    1:关    2:关    3:关    4:关    5:关    6:关
ebtables        0:关    1:关    2:关    3:关    4:关    5:关    6:关
hsqldb          0:关    1:关    2:关    3:关    4:关    5:关    6:关
ipsec           0:关    1:关    2:关    3:关    4:关    5:关    6:关
iscsi           0:关    1:关    2:关    3:开    4:开    5:开    6:关
iscsid          0:关    1:关    2:关    3:开    4:开    5:开    6:关
netconsole      0:关    1:关    2:关    3:关    4:关    5:关    6:关
network         0:关    1:关    2:关    3:关    4:关    5:关    6:关
openct          0:关    1:关    2:开    3:开    4:开    5:开    6:关
spice-vdagentd  0:关    1:关    2:关    3:关    4:关    5:开    6:关
tcsd            0:关    1:关    2:开    3:开    4:开    5:开    6:关
zfs-fuse        0:关    1:关    2:关    3:关    4:关    5:关    6:关
```

【例11.28】列出ipsec服务在各个运行级别上的运行状态。

[root@PC-LINUX ~]# chkconfig --list ipsec

注意：该输出结果只显示 SysV 服务，并不包含原生 systemd 服务。SysV 配置数据可能被原生 systemd 配置覆盖。
ipsec 0:关 1:关 2:关 3:关 4:关 5:关 6:关

【例11.29】在运行级别3，4，5上启动named服务。

[root@PC-LINUX ~]# chkconfig --level 345 named on
注意：正在将请求转发到“systemctl enable named.service”。
ln -s '/usr/lib/systemd/system/named.service' '/etc/systemd/system/multi-user.target.wants/named.service'

【例11.30】在运行级别3，4上停止named服务。

[root@PC-LINUX ~]# chkconfig --level 34 named off
注意：正在将请求转发到“systemctl disable named.service”。
rm '/etc/systemd/system/multi-user.target.wants/named.service'

【例11.31】对httpd服务设置没有选择运行级别的启动。

[root@PC-LINUX ~]# chkconfig httpd on
注意：正在将请求转发到“systemctl enable httpd.service”。
ln -s '/usr/lib/systemd/system/httpd.service' '/etc/systemd/system/multi-user.target.wants/httpd.service'
//当前httpd服务在2，3，4，5运行级别上启动

2. service命令

使用service命令可以启动或停止守护进程，service命令执行后会立即生效。

命令语法：

service [服务名] [start|stop|restart|status]

【例11.32】 启动named服务。

```
[root@PC-LINUX ~]# service named start
Redirecting to /bin/systemctl  start named.service
```

【例11.33】 查看named服务运行状态。

```
[root@PC-LINUX ~]# service named status
Redirecting to /bin/systemctl  status named.service
named.service - Berkeley Internet Name Domain (DNS)
          Loaded: loaded (/usr/lib/systemd/system/named.service; disabled)
          Active: active (running) since Thu, 30 May 2013 14:44:06 +0800; 1s ago
         Process: 4653 ExecStop=/bin/sh -c /usr/sbin/rndc stop > /dev/null 2>&1 ||
/bin/kill -TERM $MAINPID (code=exited, status=0/SUCCESS)
         Process: 4678 ExecStart=/usr/sbin/named -u named $OPTIONS
(code=exited, status=0/SUCCESS)
         Process: 4676 ExecStartPre=/usr/sbin/named-checkconf -z /etc/named.conf
(code=exited, status=0/SUCCESS)
        Main PID: 4679 (named)
          CGroup: name=systemd:/system/named.service
                  └ 4679 /usr/sbin/named -u named
```

May 30 14:44:06 PC-LINUX named[4679]: open: /etc/rndc.key: file not found
May 30 14:44:06 PC-LINUX named[4679]: couldn't add command channel ::1#953: file not found
May 30 14:44:06 PC-LINUX named[4679]: managed-keys-zone: loaded serial 3
May 30 14:44:06 PC-LINUX named[4679]: zone 0.in-addr.arpa/IN: loaded serial 0
May 30 14:44:06 PC-LINUX named[4679]: zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
May 30 14:44:06 PC-LINUX named[4679]: zone localhost/IN: loaded serial 0
May 30 14:44:06 PC-LINUX named[4679]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa/IN: loaded serial 0
May 30 14:44:06 PC-LINUX named[4679]: zone localhost.localdomain/IN: loaded serial 0
May 30 14:44:06 PC-LINUX named[4679]: all zones loaded
May 30 14:44:06 PC-LINUX named[4679]: running

【例11.34】 重启named服务。

```
[root@PC-LINUX ~]# service named restart
Redirecting to /bin/systemctl  restart named.service
```

【例11.35】 停止named服务。

```
[root@PC-LINUX ~]# service named stop
Redirecting to /bin/systemctl  stop named.service
```

11.5 实现Linux网络安全

Linux系统是一种较安全的操作系统，但是到目前为止还没有任何一种操作系统可以实现百分之百的安全。本节主要讲述提高Linux系统安全性能和Linux系统安全保护措施的一些方法。

11.5.1 提高Linux系统安全性能

就系统安全性而言，Linux系统相对于Windows系统具有更多的优势。但是，不管选择哪一种Linux发行版本，在安装完成以后都应该进行一些必要的配置，来增强它的安全性。下面介绍提高Linux系统安全性的一些方法。

1. 部署防火墙
2. 关闭不用的服务和端口
3. 严格禁止设置默认路由
4. 口令管理
5. 分区管理
6. 防范网络嗅探
7. 完整的日志管理
8. 使用安全工具软件
9. 使用保留的IP地址

10．部署Linux防病毒软件

（1）针对Linux本身防范策略

（2）针对使用Linux服务器后端的Windows系统的病毒防范策略

11．加强登录安全

表11-1　/etc/login.defs文件的字段含义

字　　段	含　　义
PASS_MAX_DAYS 90	登录密码有效期90天
PASS_MIN_DAYS 0	登录密码最短修改时间是0天，防止非法用户短期更改多次密码
PASS_MIN_LEN 8	登录密码最小长度8位
PASS_WARN_AGE 7	登录密码过期提前7天提示修改
FAIL_DELAY 10	登录错误时等待时间10s
FAILLOG_ENAB yes	登录错误记录到日志
SYSLOG_SU_ENAB yes	当限定超级用户管理日志时使用
SYSLOG_SG_ENAB yes	当限定超级用户组管理日志时使用

12．补丁问题

11.5.2 Linux系统安全保护措施

在Linux系统下可以使用本小节知识来设置安全保护措施，保证Linux系统的安全，不被其他用户攻击。

1．系统安全记录文件
2．启动和登录安全性
（1）BIOS安全
（2）用户口令
（3）默认账号
（4）口令文件
（5）限制使用su命令

3．限制NFS访问
4．登录终端设置
5．防止网络攻击
（1）阻止ping
（2）防止IP欺骗
（3）防止DOS攻击
6．安装补丁

小　　结

用户可以在Fedora Core 17系统中编辑相应的网络配置文件来完成网络配置工作，这些文件主要有/etc/sysconfig/network-scripts/ifcfg-eth0，/etc/resolv.conf，/etc/host.conf，/etc/sysconfig/network，/etc/hosts和/etc/services。

小　　结

在Linux系统中提供了大量的网络命令用于网络配置、网络测试以及网络诊断，如ifconfig，ping，netstat，traceroute，arp以及tcpdump等。

在Fedora Core 17系统中使用“网络配置”工具可以用来配置计算机的IP地址、子网掩码、网关地址、首选DNS服务器IP地址等信息。

小　　结

管理Linux系统服务的方法有很多，可以根据不同的服务、系统配置以及对Linux系统的掌握程度来决定使用哪一种管理方法。可以使用ntsysv、systemctl以及chkconfig和service命令启动或停止这些服务。其中使用ntsysv可以允许为每个运行级别配置引导时要启动的服务，对于独立服务而言，改变不会立即生效。

小　　结

就系统安全性而言，Linux系统相对于Windows系统具有更多的优势。但是，不管选择哪一种Linux发行版本，在安装完成以后都应该进行一些必要的配置，以增强它的安全性，不被其他用户攻击。