《Linux 服务器安全策略详解》

著者： 曹江华编著
ISBN 号： 978-7-121-04571-4
出版日期： 2007-07
字数：1133 千字　　页码：626　　开本：16 开

详细目录：

http://www.phei.com.cn/bookshop/bookinfo.asp?bookcode=TP045710%20&booktype=new

http://www.china-pub.com/computers/common/info.asp?id=35196

在线连载：

http://book.csdn.net/bookfiles/430/

第 1 章 Linux 网络基础与 Linux 服务器的安全威胁

本章要点

- Linux 网络基础
- Linux 的 TCP/IP 网络配置
- 分级解析对 Linux 服务器的攻击
- 开源软件网络安全概述

1.1　Linux 网络基础

1.1.1　Linux 网络结构的特点

Linux 在服务器领域已经非常成熟，其影响力日趋增大。Linux 的网络服务功能非常强大，但是由于 Linux 的桌面应用和 Windows 相比还有一定差距，除了一些 Linux 专门实验室之外，大多数企业在应用 Linux 系统时，往往是 Linux 和 Windows（或 UNIX）等操作系统共存形成的异构网络。

在一个网络系统中，操作系统的地位是非常重要的。Linux 网络操作系统以高效性和灵活性而著称。它能够在 PC 上实现全部的 UNIX 特性，具有多任务、多用户的特点。Linux 的组网能力非常强大，它的 TCP/IP 代码是最高级的。Linux 不仅提供了对当前的 TCP/IP 协议的完全支持，也包括了对下一代 Internet 协议 IPv6 的支持。Linux 内核还包括了 IP 防火墙代码、IP 防伪、IP 服务质量控制及许多安全特性。Linux 的网络实现是模仿 FreeBSD 的，它支持 FreeBSD 的带有扩展的 Sockets（套接字）和 TCP/IP 协议。它支持两个主机间的网络连接和 Sockets 通信模型，实现了两种类型的 Sockets：BSD Sockets 和 INET Sockets。它为不同的通信模型提供了两种传输协议，即不可靠的、基于消息的 UDP 传输协议和可靠的、基于流的 TCP 传输协议，并且都是在 IP 网际协议上实现的。INET Sockets 是在以上两个协议及 IP 网际协议之上实现的，它们之间的关系如图 1-1 所示。

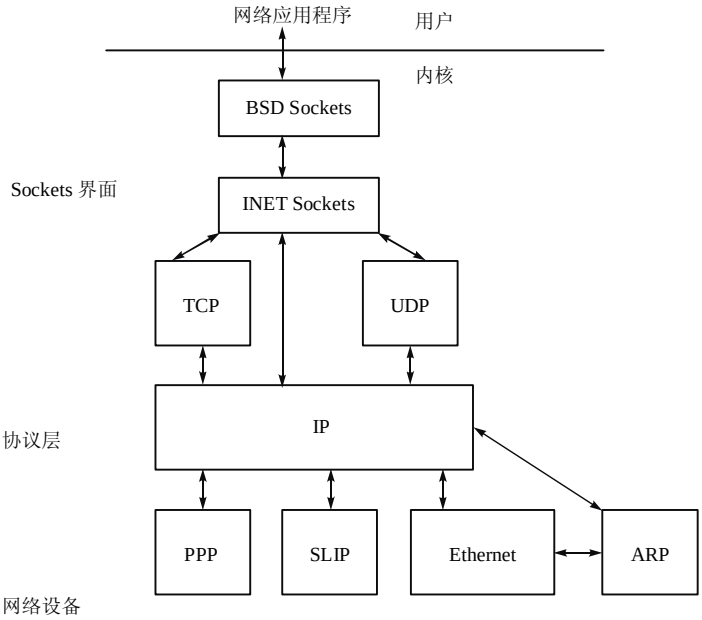

图 1-1　Linux 网络中的层

掌握 OSI 网络模型、TCP/IP 模型及相关服务对应的层次对于理解 Linux 网络服务器是非常重要的。

1.1.2　TCP/IP 四层模型和 OSI 七层模型

表 1-1 是 TCP/IP 四层模型和 OSI 七层模型对应表。我们把 OSI 七层网络模型和 Linux TCP/IP 四层概念模型对应，然后将各种网络协议归类。

表 1-1　TCP/IP 四层模型和 OSI 七层模型对应表

OSI 七层网络模型	Linux TCP/IP 四层概念模型	对应网络协议
应用层（Application）	应用层	TFTP, FTP, NFS, WAIS
表示层（Presentation）		Telnet, Rlogin, SNMP, Gopher
会话层（Session）		SMTP, DNS
传输层（Transport）	传输层	TCP, UDP
网络层（Network）	网际层	IP, ICMP, ARP, RARP, AKP, UUCP
数据链路层（Data Link）	网络接口	FDDI, Ethernet, Arpanet, PDN, SLIP, PPP
物理层（Physical）		IEEE 802.1A, IEEE 802.2 到 IEEE 802.11

1．网络接口

网络接口把数据链路层和物理层放在一起，对应 TCP/IP 概念模型的网络接口。对应的网络协议主要是：Ethernet、FDDI 和能传输 IP 数据包的任何协议。

2．网际层

网络层对应 Linux TCP/IP 概念模型的网际层，网络层协议管理离散的计算机间的数据传输，如 IP 协议为用户和远程计算机提供了信息包的传输方法，确保信息包能正确地到达目的机器。这一过程中，IP 和其他网络层的协议共同用于数据传输，如果没有使用一些监视系统进程的工具，用户是看不到在系统里的 IP 的。网络嗅探器 Sniffers 是能看到这些过程的一个装置（它可以是软件，也可以是硬件），

它能读取通过网络发送的每一个包，即能读取发生在网络层协议的任何活动，因此网络嗅探器 Sniffers 会对安全造成威胁。重要的网络层协议包括 ARP（地址解析协议）、ICMP（Internet 控制消息协议）和 IP 协议（网际协议）等。

3．传输层

传输层对应 Linux TCP/IP 概念模型的传输层。传输层提供应用程序间的通信。其功能包括：格式化信息流；提供可靠传输。为实现后者，传输层协议规定接收端必须发回确认信息，如果分组丢失，必须重新发送。传输层包括 TCP（Transmission Control Protocol，传输控制协议）和 UDP（User Datagram Protocol，用户数据报协议），它们是传输层中最主要的协议。TCP 建立在 IP 之上，定义了网络上程序到程序的数据传输格式和规则，提供了 IP 数据包的传输确认、丢失数据包的重新请求、将收到的数据包按照它们的发送次序重新装配的机制。TCP 协议是面向连接的协议，类似于打电话，在开始传输数据之前，必须先建立明确的连接。UDP 也建立在 IP 之上，但它是一种无连接协议，两台计算机之间的传输类似于传递邮件：消息从一台计算机发送到另一台计算机，两者之间没有明确的连接。UDP 不保证数据的传输，也不提供重新排列次序或重新请求的功能，所以说它是不可靠的。虽然 UDP 的不可靠性限制了它的应用场合，但它比 TCP 具有更好的传输效率。

4．应用层

应用层、表示层和会话层对应 Linux TCP/IP 概念模型中的应用层。应用层位于协议栈的顶端，它的主要任务是应用。一般是可见的，如利用 FTP（文件传输协议）传输一个文件，请求一个和目标计算机的连接，在传输文件的过程中，用户和远程计算机交换的一部分是能看到的。常见的应用层协议有：HTTP，FTP，Telnet，SMTP 和 Gopher 等。应用层是 Linux 网络设定最关键的一层。Linux 服务器的配置文档主要针对应用层中的协议。TCP/IP 模型各个层次的功能和协议如表 1-2 所示。

表 1-2　TCP/IP 模型各个层次的功能和协议

层次名称	功　能	协　议
网络接口 （Host-to-Net Layer）	负责实际数据的传输，对应 OSI 参考模型的下两层	HDLC（高级链路控制协议） PPP（点对点协议） SLIP（串行线路接口协议）
网际层 （Inter-network Layer）	负责网络间的寻址 数据传输，对应 OSI 参考模型的第三层	IP（网际协议） ICMP（网际控制消息协议） ARP（地址解析协议） RARP（反向地址解析协议）
传输层 （Transport Layer）	负责提供可靠的传输服务，对应 OSI 参考模型的第四层	TCP（控制传输协议） UDP（用户数据报协议）
应用层 （Application Layer）	负责实现一切与应用程序相关的功能，对应 OSI 参考模型的上三层	FTP（文件传输协议） HTTP（超文本传输协议） DNS（域名服务器协议） SMTP（简单邮件传输协议） NFS（网络文件系统协议）

说明　TCP/IP 与 OSI 最大的不同在于 OSI 是一个理论上的网络通信模型，而 TCP/IP 则是实际运行的网络协议。

1.1.3 TCP/IP 提供的主要用户应用程序

1．Telnet 程序

Telnet 程序提供远程登录功能。

2．文件传输协议

文件传输协议（FTP）允许用户将一个系统上的文件复制到另一个系统上。

3．简单邮件传输协议

简单邮件传输协议（SMTP）用于传输电子邮件。

4．Kerberos 协议

Kerberos 是一个受到广泛支持的安全性协议。

5．域名服务器协议

域名服务器协议（DNS）能使一台设备具有的普通名字转换成某个特定的网络地址。

6．简单网络管理协议

简单网络管理协议（SNMP）把用户数据报协议（UDP）作为传输机制，它使用和 TCP/IP 不同的术语，TCP/IP 用客户端和服务器，而 SNMP 用管理器（Manager）和代理（Agent），代理提供设备信息，而管理器管理网络通信。

7．网络文件系统协议

网络文件系统协议（NFS）是由 SUN Microsystems 公司开发的一套协议，可使多台计算机能透明地访问彼此的目录。

8．远程过程调用

远程过程调用（RPC）是使应用软件能与另一台计算机（服务器）通信的一些函数。

9．普通文件传输协议

普通文件传输协议（TFTP）是一种缺乏任何安全性的、非常简单落后的文件传输协议。

10．传输控制协议

传输控制协议（TCP/IP 中的 TCP 部分）是一种数据可靠传输的通信协议。

11．网际协议

网际协议（IP）负责在网络上传输由 TCP/UDP 装配的数据包。

12．网际控制消息协议

网际控制消息协议负责根据网络上设备的状态发出和检查消息，它可以将某台设备的故障通知到其他设备。

1.1.4 端口号分配

TCP 和 UDP 采用 16bit 的端口号来识别应用程序。那么这些端口号是如何选择的呢？

服务器一般都是通过知名端口号来识别的。例如，对于 TCP/IP 实现来说，每个 FTP 服务器的 TCP 端口号都是 21，每个 Telnet 服务器的 TCP 端口号都是 23，每个 TFTP（普通文件传输协议）服务器的 UDP 端口号都是 69。任何 TCP/IP 实现所提供的服务都用知名的 1~1 023 之间的端口号。这些知名端口号由 Internet 号分配机构（Internet Assigned Numbers Authority, IANA）来管理。到 1992 年为止，知名端口号介于 1~255 之间。256~1 023 之间的端口号通常都是由 UNIX 系统占用，以提供一些特定的 UNIX 服务，也就是说，提供一些只有 UNIX 系统才有的，而其他操作系统可能不提供的服务。现在 IANA 管理 1~1 023 之间所有的端口号。

Internet 扩展服务与 UNIX 特定服务之间的一个差别就是 telnet 和 rlogin，它们二者都允许通过计算机网络登录到其他主机上。telnet 是采用端口号为 23 的 TCP/IP 标准，且几乎可以在所有操作系统上进行实现。相反，rlogin 最开始时只是为 UNIX 系统设计的（尽管许多非 UNIX 系统现在也提供该服务），因此在 20 世纪 80 年代初，它的端口号为 513，客户端通常对它所使用的端口号并不关心，只须保证该端口号在本机上是唯一的即可。客户端口号又称做临时端口号（即存在时间很短暂），这是因为它通常只是在用户运行该客户程序时才存在，而服务器则只要主机开着，其服务就运行。

大多数 TCP/IP 实现给临时端口分配 1 024~5 000 之间的端口号。大于 5 000 的端口号是为其他服务器预留的（Internet 上并不常用的服务）。我们可以在后面看见许多给临时端口分配端口号的例子。大多数 Linux 系统的文件/etc/services 都包含了人们熟知的端口号。为了找到 telnet 服务，可以运行以下语句：“grep telnet /etc/services”。表 1-3 是一些常用 TCP 服务和端口。

表 1-3　常用 TCP 服务和端口

TCP 端口	服　务　名	功　　能
7	echo	echo 字符（用于测试）
9	discard	丢弃字符串（用于测试）
13	daytime	日期服务
19	chargen	字符生成器
21	ftp	文件传输协议（FTP）
22	ssh	安全 shell（虚拟终端或文件传输）
23	telnet	远程登录
25	smtp	电子邮件
37	time	时间服务
42	nameserve	TCP 名字服务
43	whois	NIC whois 服务
53	domain	域名服务（DNS）
79	finger	用户信息
80	http	WWW（万维网）
110	pop3	邮局协议 3（POP3）
111	sunrpc	SUN 的远程过程调用（RPC）
113	auth	远程用户名认证服务
119	nntp	网络新闻传输协议（NNTP）
143	imap	交互式邮件访问协议
443	https	用 SSL 加密的 HTTP
512	exec	在远程 UNIX 主机上执行命令
513	login	登录到远程 UNIX 主机（rlogin）
514	shell	从远程 UNIX 主机获得 shell（rsh）

续表

TCP 端口	服 务 名	功 能
515	printer	远程打印
1080	socks	SOCKS 应用代理服务
2049	NFS	TCP 之上的 NFS（NFS over TCP）
6000~6001	X	X Window 系统

UDP 为运行于同一台或不同机器之上的两个或多个程序之间传输数据包提供了简单的、不可靠的连接。“不可靠”意味着操作系统不保证每个发出的包都能到达，也不保证包能够按序到达。不过 UDP 是尽力传输的，在 LAN 中 UDP 通常能达到 100%的可靠性。UDP 的优点在于它比 TCP 的开销少，较少的开销使得基于 UDP 的服务可以用 TCP 10 倍的吞吐量传输数据。

UDP 主要用于 SUN 的 NFS、NIS、主机名解析和传输路由信息。对于有些服务而言，偶然丢失一个包并不会带来太大的负面影响，因为它们会周期性地请求一个新包，或者那些包本身并不是很重要。这些服务包括 who、talk 和一些时间服务。表 1-4 是一些常用 UDP 服务和端口。

表 1-4 常见 UDP 服务和端口

UDP 端口	服 务 名	功 能
7	echo	在另一个数据包中返回用户的数据
9	discard	什么也不做
13	daytime	返回日期
19	chargen	字符生成器
37	time	返回时间
53	domain	域名服务（DNS）
69	tftp	普通文件传输协议
111	sunrpc	SUN 的远程过程调用（RPC）
123	ntp	网络时间协议（Network Time Protocol，NTP）
161	snmp	简单网络管理协议
512	biff	新邮件提示
513	who	收集关于用户登录到同一子网的其他机器的广播
514	syslog	系统日志工具
517	talk	发送 talk 请求
518	ntalk	一个“新”的 talk 请求
520	route	路由信息协议
533	netwall	写每个用户的终端
2049	NFS	网络文件系统协议（NFS）

1.2 Linux 的 TCP/IP 网络配置

Linux 从一开始就是为网络而设计的。它内置了以前仅在高端企业产品中才可见到的成熟功能。然而，尽管拥有这些强大的能力，Linux 网络的配置却远没有 Windows 网络的配置复杂。诸如 Webmin、redhat-config-network 和 YaST 允许执行图形化的配置；诸如 ifconfig 和 route 允许通过控制台或脚本查看和修改网络参数；诸如 netstat 允许查看单独的网络连接，并显示它们与运行着的进程的关系。

1.2.1　Linux 的 TCP/IP 网络配置文件

除非另行指定，Red Hat Linux 系统中大多数配置文件都在 /etc 目录中。配置文件如表 1-5 所示。

表 1-5　配置文件

配置文件名称	功　　能
/etc/gated.conf	gated 的配置，只能被 gated 守护进程所使用
/etc/gated.version	gated 守护进程的版本号
/etc/gateway	由 routed 守护进程可选择地使用
/etc/networks	列举机器所连接的网络中可以访问的网络名和网络地址。通过路由命令使用，允许使用网络名称
/etc/protocols	列举当前可用的协议，请参阅网络管理员指南和联机帮助页
/etc/resolv.conf	在程序请求“解析”一个 IP 地址时，告诉内核应该查询哪个名称服务器
/etc/rpc	包含 RPC 指令/规则，这些指令/规则可以在 NFS 调用、远程文件系统安装等中使用
/etc/exports	要导出的网络文件系统（NFS）和对它的权限
/etc/services	将网络服务名转换为端口号/协议，由 inetd、telnet、tcpdump 和一些其他程序读取，有一些 C 访问例程
/etc/xinetd.conf	xinetd 的配置文件，请参阅 xinetd 联机帮助页。包含每个网络服务的条目，inetd 必须为这些网络服务控制守护进程或其他服务。注意，服务将会运行，但在/etc/services 中将它们注释掉了，这样即使这些服务在运行也将不可用
/etc/hostname	该文件包含了系统的主机名称，包括完全的域名，例如 www.linuxaid.com.cn
/etc/host.conf	该文件指定如何解析主机名。Linux 通过解析器来获得主机名对应的 IP 地址
/etc/sysconfig/network	指出 NETWORKING=yes 或 no，由 rc.sysinit 读取
/etc/sysconfig/network-scripts/if*	Red Hat 网络配置脚本
/etc/hosts	机器启动时，在查询 DNS 以前，机器需要查询一些主机名与 IP 地址的匹配信息，这些匹配信息存放在 /etc/hosts 文件中。在没有域名服务器情况下，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

1.2.2　网络配置工具

在安装 Linux 发行版本时，需要配置网络。你或许已经有一个来自初始配置的活动 eth0，这个配置对于当前的使用也许足够，但是随着时间的推移你可能需要做出更改。下面将介绍与 IP 网络相关的不同配置项，以及使用这些配置项的文件和工具。

1．手动修改配置文件

手动配置是最直接的方式，熟练的 Linux 用户在平时维护系统的时候更喜欢使用手工配置，因为手工配置有很多优点：

① 熟悉命令之后，手工配置更快速，并且不需要重新启动；
② 能够使用配置命令的高级特性；
③ 更容易维护配置文件，找出系统故障；
④ 能更深刻地了解系统配置是如何进行的。

2．使用 Linux 命令

虽然 Linux 桌面应用发展很快，但是命令在 Linux 中依然有很强的生命力。Linux 是一个由命令行组成的操作系统，其精髓在于命令行，无论图形界面发展到什么水平这个原理是不会变的。Linux 网络设备操作命令包括 ifconfig, ip, ping, netstat, route, ip, arp, hostname 和 arpwatch。

3．Webmin

Webmin 在 Networking 下的 Network Configuration 中提供了一组网络配置工具。你可以配置单独的接口，并调整它们的当前设置或已保存的设置。还可以配置路由和网关、DNS 客户端设置，以及本地主机地址。编辑好所有的配置之后，可以单击“Apply Configuration”来应用它们，不必重新启动系统。

4．不同发行版本中的工具

每个发行版本都有它自己用于配置网络设置的工具。应该参考特定发行版本的文档来确定要使用的工具。每种工具都提供了与 Webmin 工具基本相同的配置选项。其中有些版本可能提供特定于该发行版本的选项。Red Hat Linux 3/4 使用 system-config-network 工具（如图 1-2 所示），SuSE Linux 使用 YaST 工具（如图 1-3 所示）。

图 1-2　Red Hat Linux 3/4 使用 system-config-network 工具

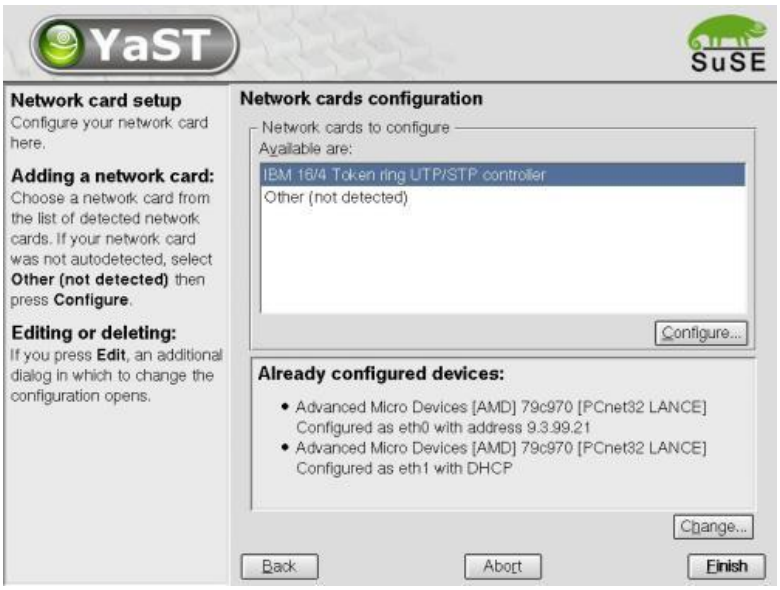

图 1-3　SuSE Linux 使用 YaST 工具

1.2.3　配置网络接口

1．网卡的选择

一般来说，2.4 版本以后的 Linux 可以支持的网卡芯片组数量已经很完备了，包括著名厂商，如 Intel，以及使用广泛的 RealTek、Via 等网卡芯片都已经被支持，所以使用者可以很轻松地设定好他们的网卡。但是由于 Linux 发行版本众多（目前超过 188 个），使用前最好先查看 Linux 发行版本的文档。以 RHEL 4.0 为例，这个设备列表在 Ethernet-HOWTO 文档中。另外最直接的方法是查看目录/lib/modules/ release/kernel/drivers/net，其中 release 是内核版本，可以使用命令“uname-r”获得，对于 RHEL 4.0 内核版本是 2.6.9-5EL。

```
#ls /lib/modules/2.6.9-5EL/kernel/drivers/net/
3c59x.ko    bonding     fealnx.ko     netdump.ko     pppox.ko
starfire.ko   tun.ko      8139cp.ko   dl2k.ko     forcedeth.ko
ns83820.ko     ppp_synctty.ko  sungem.ko     typhoon.ko8139too.ko
dummy.ko    hp100.ko      pcmcia         r8169.ko       sungem_phy.ko
via-rhine.ko82596.ko    e1000       ixgb          pcnet32.ko     s2io.ko
sunhme.ko     via-velocity.ko8390.ko     e100.ko     mii.ko
ppp_async.ko   sis900.ko      tg3.ko        wirelessacenic.ko
eepro100.ko  natsemi.ko    ppp_deflate.ko  sk98lin      tlan.koamd8111e.ko
epic100.ko   ne2k-pci.ko   ppp_generic.ko  slhc.ko        tokenringb44.ko
ethertap.ko  netconsole.ko  pppoe.ko  smc9194.ko     tulip
```

可以看到这个目录列出所有 Linux 内核支持的网络设备驱动程序。其中大部分是以太网卡（8139、3COM、Intel），也有一些是其他类型的设备。对于初学者应当尽量选择目录中已经列出的网卡。

注意　以.o 后缀结束的文件就是驱动程序，而没有后缀的是驱动程序目录。

2．检查网卡是否加载

驱动硬件是操作系统最基本的功能，操作系统通过各种驱动程序来驾驭硬件设备，和 Windows 系统不同，Linux 内核目前采用可加载的模块化设计（LKMs Loadable Kernel Modules），就是将最基本的核心代码编译在内核中，网卡驱动程序是作为内核模块动态加载的，可以使用命令“lsmod”查看加载情况：

```
## lsmod
Module                  Size  Used by
dm_mod                 54741  0
button                  6481  0
battery                 8901  0
ac                      4805  0
md5                     4033  1
joydev                 10241  0
uhci_hcd               31065  0
ehci_hcd               30917  0
snd_via82xx            26437  0
snd_ac97_codec         63889  1 snd_via82xx
snd_pcm_oss            49017  0
soundcore               9889  1 snd
tulip                  45025  1
```

```
via_rhine           23113  2
mii                  4673  1 via_rhine
ext3               116809  2
jbd                 71257  1 ext3
```

对每行而言，第一列是模块名称；第二列是模块大小；第三列是调用数。调用数后面的信息对每个模块而言都有所不同。如果（unused）被列在某模块的那行中，说明该模块当前未被使用。如果（autoclean）被列在某模块的那行中，说明该模块可以被 rmmod-a 命令自动清除，当这个命令被执行后，所有自从上次被自动清除后，未被使用的且标记了“autoclean”的模块都会被卸载。从以上粗体字符可以看到 Linux 计算机中两块网卡模块 tulip 和 via_rhine 已经加载。对应的网卡商业型号分别是：

– tulip　Lite-On Communications Inc LNE100TX [Linksys EtherFast 10/100]。

– via_rhine　Via VT6102[Rhine-II] 常见主板集成网卡。

如果没有检测到硬件，用硬件检测程序 kuduz 检测网卡，它和 Windows 中添加新硬件功能差不多。kudzu 程序是通过查看/usr/share/hwdata/目录下的文件识别各种硬件设备的。如果内核支持该硬件，并且有该驱动程序就可自动装载。首先需要说明的是，Linux 下对网卡的支持往往是针对芯片的，所以对某些不是很著名的网卡，需要知道它的芯片型号以配置 Linux，比如 Top link 网卡，就不存在 Linux 的驱动，但是因为它与 NE2000 兼容，所以把它当做 NE2000 就可以在 Linux 下使用了。所以当你有一块网卡不能使用，在未找到 Linux 的驱动程序之前，一定搞清楚这个网卡使用的是什么芯片，跟谁兼容，比如 3c509，NE2000 等。这样的型号一般都在网卡上最大的一块芯片上印着，抄下来就是了。对于 ISA 接口的 NE2000 卡，首先需要将网卡设定为 Jumpless 模式。目前很多网卡默认的都是 PnP 模式，这在 Windows 下的确能减少很多麻烦，但是 Linux 不支持，所以 Linux 下必须是 Jumpless 模式。一般所有网卡都带有驱动盘和在 DOS 下可执行的一个设定程序，用该程序将网卡设为 Jumpless。对于 PCI 网卡，可以使用 lspci 命令来查看。在显示的列表中找到“Ethernet Controller”，记下厂商和型号，然后使用 modprobe 尝试加载正确的模块，比如 modprobe 3c509。如果出现错误，说明该模块不存在。这时候你应该找到正确的模块并且重新编译。

如果你使用的是比较罕见的一些网卡，或者是 Linux 核心支持不够的网卡，以致在安装 Linux 时无法检测到网卡，那也不用担心，我们可以使用较为简单的核心模块编译来支持这块网卡。下面以 3COM 的 3CR990-TX-97 网卡为例（一款具有安全特性的网卡）看看如何进行模块编译。首先在其网站 http://www.3com.com/infodeli/tools/nic/linuxdownload.htm 下载适合你使用内核版本的相关驱动程序，这里以 2.4 内核为例。

```
#wegt http://www.3com.com/infodeli/tools/nic/3c990-1.0.0a.tar.gz
```

另外在开始编译核心模块之前，因为驱动程序需要配合核心来编译，所以会使用到 kernel source 或者是 kernel header 的数据，此外，也需要编译器（compiler）的帮助，因此，先确定你的 Linux 系统当中已经存在下列软件：kernel-source, kernel, gcc, make。

```
#tar zxvf 3c990-1.0.0a.tar.gz
#make
```

此时会产生 3c990.o 驱动模块，然后使用命令复制到相应位置，查看加载是否正常。

```
#modprobe 3c990
#cp 3c990.o  /lib/modules/2.4.20-8/kernel/drivers/net
# depmod –a
```

然后使用 lsmod 命令检查加载情况，如果一切正常，可以让系统启动时自动加载该模块：

```
#echo “alias eth0 3c990”>> /etc/modules.conf
```

3．为新网卡设定 IP 地址

Linux 网络设备在配置时被赋予别名，该别名由一个描述性的缩略词和一个编号组成。某种类型的第一个设备的编号为 0，其他设备依次被编号为 1，2，3 等。但是网卡并不是作为裸设备出现在/dev 目录下，而是存在内存中。eth0、eth1 是以太网卡接口，它们用于大多数的以太网卡，包括许多并行端口以太网卡。本节主要讨论这类网卡。为 Linux 以太网卡设定 IP 地址的方式非常灵活，你可以选择适合你工作情况的方法。

（1）使用 ip 或 ifconfig 命令

ifconfig 命令是最重要的 Linux 网络命令，它最主要的用途是设定、修改网卡的 IP 地址：

```
#ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

默认情况下，ifconfig 显示活动的网络设备。给这个命令添加一个“-a”开关就能看到所有设备。但是 ifconfig 命令设置完网络设备的 IP 地址，当系统重新启动后设置会自动失效，所以它主要用于网卡状态调试。假设你要建立一个临时的网络配置以供测试，你可以使用发行版本中的工具来编辑配置，但是需要注意，在完成测试之后，将所有设置恢复回去。通过使用 ifconfig 命令，我们无须影响已保存的设置，就能够快速地配置网卡。

ip 命令是 iproute2 软件包中的一个强大的网络配置工具，它能够替代一些传统的网络管理工具，例如 ifconfig、route 等。现在，绝大多数 Linux 发行版和绝大多数 UNIX 都使用古老的 arp、ifconfig 和 route 命令。虽然这些工具能够工作，但它们在 Linux2.2 和更高版本的 Linux 内核上显得有些落伍。使用 iproute2 前，你应该确认已经安装了这个工具。这个包的名字在 Red Hat Linux 9.0 叫做“iproute2”，也可以在 ftp://ftp.inr.ac.ru/ip-routing/下载源代码安装。

在以太网接口 eth0 上增加一个地址 10.0.0.1，掩码长度为 24 位，标准广播地址，标签为“eth0:Alias：”

```
#ip addr add 10.0.0.1/24 brd + dev eth0 label eth0:Alias
```

（2）使用 netconfig 命令

netconfig 命令可以设置网络设备的 IP 地址，netconfig 命令可以永久保存设置。使用方法是：“netconfig ethX”。使用命令“netconfig eth0”后，会在命令行下弹出一个对话框，这时即可进行设定，如图 1-4 所示。

图 1-4　netconfig 配置界面

设定结束后，用“Tab”键选择“OK”，即可保存设置并且退出，然后使用命令“service network restart”激活，即可生效。

（3）使用 neat 命令

使用 neat 命令需要配置好 X Window 系统，在命令行下运行 neat 命令后，添加 IP 地址和其他相关参数后保存设置，重新启动网络和网络服务或计算机，如图 1-5 所示。

图 1-5　图形界面添加 IP 地址

另外，neat 命令还有一个等价命令“redhat-config-network”，二者完全相同。neat 和 redhat-config-config 命令可以永久保存设置。

（4）修改 TCP/IP 网络配置文件

除非另行指定，Red Hat Linux 网卡相关的 TCP/IP 网络配置文件是/etc/sysconfig/network-scripts/ifcfg-eth×，其中×从 0 开始，第一个以太网配置文件即/etc/sysconfig/network-scripts/ifcfg-eth0。使用 vi 编辑器可以修改这个文件，也可以修改网卡 IP 地址。

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0                //设定网卡的名称，要跟文件名称对应
ONBOOT=yes                 //是否在开机的时候启动网卡
BOOTPROTO=static           //启动的时候 IP 取得的协议，这里是固定的，如果是动态主机的话，要改成 dhcp
IPADDR=192.168.1.2         //IP 地址
NETMASK=255.255.255.0      //子网掩码
NETWORK=192.168.1.0        //该网段的第一个 IP
BROADCAST=192.168.1.255    //最后一个同网段的广播地址
GATEWAY=192.168.1.2        //网关地址#
#GATEWAYDEV=eth0
```

存盘后使用命令“service network restart”激活即可生效。该方法同样可以永久保存设置。

（5）为网卡添加 IPv6 地址

和 Windows 操作系统相比，Linux 对 IPv6 的支持更好，最早的支持 IPv6 的 Linux 内核是 2.2。一般基于 2.4 内核的 Linux 发行版本都可以直接使用 IPv6，使用前要看系统的 IPv6 模块是否被加载，如

果没有被加载可以使用命令手工加载，这需要超级用户的权限。然后使用命令检测，如果显示 IPv6 地址（inet6 addr：fe80::200:e8ff:fea0:2586/64），证明 IPv6 已经加载。

```
# modprobe IPv6；#ifconfig –a
```

如果希望 Linux 系统启动时自动加载 IPv6 模块，可以在配置文件/etc/modules.conf 中加入一行：

```
alias net-pf-10 ipv6 # automatically load IPv6 module on demand
```

4．调整网卡工作模式

现在的网卡大多是自适应工作模式，在配置网卡参数时，我们很少考虑它的工作模式，有时发现一些网卡模块已经加载，但是在某些模式工作不稳定。如一块 XXX 品牌的杂牌 RTL-8139C 芯片 10/100 自适应网卡，在 100M 全双工状态下极其不稳定（在 Qcheck 的 TCP 和 UDP 的测试过程中，数据包遗失率 9.12%）。在 Linux 环境下，我们可以使用系统自带的工具 mii-tool 命令来配置网卡工作模式。显示 Linux 服务器网卡支持的所有以太网卡类型，使用命令：

```
# mii-tool -v
eth0: negotiated 100baseTx-FD, link ok
product info: vendor 00:00:00, model 0 rev 0
basic mode:   autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
advertising:  100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
```

从以上信息中可以看出，这块网卡工作在 100M 全双工自适应模式下，“100baseTx-FD”表示 100M Full Duplex。这里可以强制网卡工作在 100M 半双工模式下，输入命令：

```
#mii-tool –F 100baseTx-HD eth0
```

然后恢复网卡的自适应工作模式，输入命令：

```
#mii-tool –r eth0
```

另外，在路由器、交换机、代理服务器等通信量比较大的关键设备上，应该为它指定正确的工作模式，这样可以提高通信效率。

5．DHCP 客户端的网卡设定

DHCP 是动态主机配置协议，这个协议用于向计算机自动提供 IP 地址、子网掩码和路由信息。当设备接入这个局域网时，它们会向 DHCP 服务器请求一个 IP 地址。然后 DHCP 服务器为每个请求的设备分配一个地址，直到分配完该范围内的所有 IP 地址为止。已经分配的 IP 地址必须定时地延长借用期。这个延期的过程称做 leasing，确保了当客户机设备在正常释放 IP 地址之前，突然从网络断开时被分配的地址可以归还给服务器。Linux 下配置 DHCP 客户端有两种方法：图形界面和手工配置。使用图形界面可以使用 neat 命令界面（如图 1-5 所示），选中“自动获得 IP 地址设置使用 DHCP”即可。

选择手工配置 DHCP 的客户，需要修改/etc/sysconfig/network 文件来启用联网，并修改/etc/sysconfig/network-scripts 目录中每个网络设备的配置文件。在该目录中，每个设备都有一个叫做 ifcfg-ethX 的配置文件，ethX 是网络设备的名称，如 eth0 等。如果你想在引导时启动联网，NETWORKING 变量必须被设为 yes。除此之外，/etc/sysconfig/network 文件应该包含以下行：

```
NETWORKING=yes
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

6．在 Linux 下安装无线网卡

随着 Linux 网络技术的快速增长，硬件厂商大大提高了硬件产品对 Linux 的技术支持。使得 Linux 支持的无线网卡的数量在过去的一两年里增长显著。一旦在计算机中安装好了无线局域网卡，首先要做的就是安装驱动来让网卡工作。无线网卡实现了 IEEE 802.11 系列协议中的一种或多种的物理层（PHY）和媒质访问控制子层（MAC）的功能，而驱动是用来控制无线网卡，向上提供与以太网一致的接口和其他一些无线局域网特定的管理接口的。对于不同厂商的不同网卡，还没有一种统一的方法可以驱动所有的网卡。首先必须确保内核配置中启动了无线局域网，如果没有无线局域网支持，你应该重新配置、编译内核来启动“Wireless LAN (non-hamradio) Drivers and Wireless Extensions”。

设置无线网络相关步骤如下。

（1）用“iwconfig”命令来显示无线网卡（eth0、eth1）的信息。在以下步骤中，用 ethX 表示无线网卡的名称。

（2）设置无线网卡的操作模式为 Managed：

```
#iwconfig ethX mode Managed
```

（3）如果采用了 WEP 加密，需要设置 WEP 密码：

```
#iwconfig ethX key password XXXXXX
```

对应 40 位和 128 位加密，password 分别为 6 位和 10 位的十六进制数字。

（4）设置 SSID，其中 ESSID 为无线接入（Access Point）的 SSID：

```
# iwconfig ethX essid ESSID
```

（5）创建/etc/sysconfig/network/ifcfg-ethX 配置文件，使得机器每次启动时，无线网卡都会自动获得 IP 地址。该文件内容如下：

```
BOOTPROTO='dhcp'
MTU=''
REMOTE_IPADDR=''
STARTMODE='onboot'
UNIQUE=''
```

（6）启动无线网卡：

```
#ifconfig ethX up
```

1.3 分级解析对 Linux 服务器的攻击

随着人们对安全问题的日益重视，网络安全已经不仅仅是技术问题，而是一个社会问题。企业应当提高对网络安全的重视，不应被各种商业宣传所迷惑，认为安装了防火墙、认证授权和入侵检测系统就可以保护网络免受各种攻击。实际上，并没有绝对安全的网络，也没有“无坚不摧”的安全解决方案。从辩证法的角度来说，安全是相对的。如果一味地只依赖技术工具，那就会越来越被动；只有运用社会和法律手段打击网络犯罪，才能更加有效。我国对于打击网络犯罪已经有了明确的司法规定，

遗憾的是还没有得到大多数企业的重视，这也是本节的写作目的。

下面不仅从技术的角度解析攻击，还从社会的角度分析攻击者的特征、攻击原因、攻击目标等，攻击和安全防护是矛与盾的关系。安全与反安全之间就是一场长期战斗，了解攻击者是非常重要的。更重要的是根据攻击级别提出解决方案。

攻击是一种旨在妨碍、损害、削弱、破坏服务器安全的未授权行为。攻击的范围可以从服务拒绝直至完全危害和破坏服务器。

1.3.1　攻击者使用什么操作系统

攻击者使用的操作系统种类非常广泛。Macintosh 是很少使用的平台，因为可用于 Macintosh 操作系统的工具不多，移植所需工具相当麻烦。Windows NT/2000/XP 和 UNIX 是使用最多的平台，Linux 也比较常见。同时可以看到越来越多的攻击者正在使用 FreeBSD 或 NetBSD。

1.3.2　典型的攻击者有什么特征

（1）能用 C、C++或 Perl 编写程序。大多数原始安全工具都是用这些语言中的一种或几种编写的，攻击者必须能够解释、编译和执行这些代码。更高明一些的攻击者则能够将那些不是专为特定平台编写的代码移植到自己的平台上。同样，他们也开发新的代码模块，以扩展如 SATAN 的工具（这些程序允许用户将编写的新工具集成进去）。不过随着 J2EE 平台流行，Java 成为新的工具。

（2）深入掌握 TCP/IP 知识。攻击者必须了解 Internet 是如何工作的。攻击者必须对 TCP/IP 的原始代码有所了解，如 IP 的组成、帧的封装步骤等。

（3）每周使用 Internet 多于 72 小时，攻击者不是临时用户。他们不仅了解自己的机器，而且对网络也了如指掌。攻击者必须有丰富的网络使用经验。

（4）至少熟知三种操作系统，其中一种操作系统毫无疑问是 UNIX 或 Linux。

（5）大多数攻击者是（或曾经是）系统管理员或开发人员，具有开发客户服务器应用的经验。

1.3.3　攻击者典型的目标是什么

攻击者因不同原因而攻击不同类型的网络。我们经常能从媒体上了解到一些大公司或政府的网站遭到攻击。不过，实际上攻击者典型的目标大多是小型网络。防火墙的使用和维护费用昂贵且需要技术支持，小网络不可能用或只能用一些低级产品。攻击大公司、政府的网站会造成比较大的影响。

1.3.4　实施攻击的原因是什么

（1）恶意——他可能是某个公司的心怀不满的雇员，或许你曾在某个 Usenet 组激怒了他。

（2）娱乐——或许你曾经夸耀过你的系统的安全性，告诉别人它是如何坚不可摧，这些都是攻击者无法抗拒的挑战。

（3）获利——有人付给攻击者报酬，让他关掉某台机器或获取某公司的商业机密。

（4）好奇——许多攻击者纯粹由于好奇心的驱使，想享受一下攻击过程。

（5）政治——政治原因占攻击原因的很小比例（但是很重要的一种），他们搜寻杂志、新闻刊物中特别的论点，他们通过攻击来表达自己的政治观点和世界观。

1.3.5 攻击级别

系统攻击有许多种类，本节从攻击级别的角度进行说明。图 1-6 显示了攻击的 6 个等级，每一层代表一个进入目标网络的深度，我们称之为敏感级（Levels of sensitivity），箭头与层次相连的点标志了对应于每一破译技术的危险程度，我们将它称为攻击状态（States of attack）。

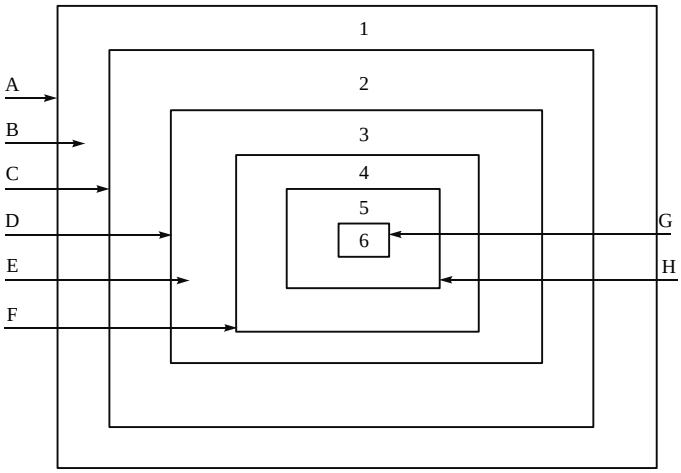

图 1-6　攻击的 6 个等级

1．级别 1

– A 箭头：表示邮件炸弹的攻击。

– B 箭头：表示简单拒绝服务攻击。

在级别 1 范围内的攻击基本上互不相关。包括拒绝服务攻击和邮件炸弹，这些攻击一般比较好制止，这是因为这些攻击是以垃圾信息方式进行的。在大多数情况下，只须应用排除模式设置就可以解决这个问题。拒绝服务攻击包括：简单拒绝服务攻击、分布拒绝服务攻击、DNS 分布拒绝服务攻击和 FTP 攻击。

对于这四种攻击可以采用以下措施：

– 关闭不必要的服务。

– 限制同时打开的 SYN 半连接数目。

– 缩短 SYN 半连接的 time out 时间。

– 及时更新系统补丁。

拒绝服务攻击经常发生，解决此问题的最佳方法就是在 inetd.sec 文件 DENY 清单中加入入侵者源主机/网络名阻止入侵行为，除屏蔽网络连接外，还没有一种主动性的方法可以避免这种攻击。不过需要注意的是，如果证实了一次拒绝服务攻击，应该检查系统是否可能遭受其他攻击，拒绝服务攻击常常是电子欺骗的先行者（甚至是组成部分）。如果观察到某台机器特定端口上的一次全面的 Flooding 攻击，请观测这个端口，弄清这个端口是干什么用的，检查它限制什么服务。如果那种服务是内部系统的组成部分，那么要特别小心。那些貌似拒绝服务的攻击，事实上就是突破网络安全的开始。通常情况下，拒绝服务攻击会持续很长一段时间。

如果是同步 Flooding 攻击，这里有一些识别攻击者的方法。攻击者在每一次实施 ping 时，向目标报出了他的 IP 地址。虽然没有给出攻击者的 E-mail 地址，但我们可以追踪其最终源（注意，追踪程序将揭示攻击者出发的真实网络地址，这通常是反向追踪程序查找的最后一项内容）。

大多数拒绝服务攻击导致相对较低的危险，即便是那些可能导致重启的攻击也仅仅是暂时性的问题。这类攻击在很大程度上不同于那些想获取网络控制的攻击。

邮件炸弹的攻击也叫邮件水灾攻击，发生在当许多邮件被发送至一个目标，发送代理人被覆盖时，邮件水灾会破坏其他交流程序的稳定。用邮件来使一个系统蒙受灾难是残酷的，但却是有效的，攻击者的目的就是要破坏邮件服务器。诱发邮件水灾攻击的有趣方法之一是利用一些邮件申请的自动反应功能。一旦黑客发现对两个不同的系统能做出活跃的、自动的应答时，他就能指使一个邮件发送到另一个。因为两者都是对每个信息做出自动应答，他们制造了一个信息回馈孔，这会比其他系统收集到更多的邮件。至于邮件水灾，通常很容易追查到攻击者。此外，bozo files（kill 文件）和排除模式配置基本上能阻止这些攻击。

2．级别 2 和级别 3

– C 箭头：表示本地用户获得非授权访问。

– D 箭头：表示本地用户获得他们不该拥有的文件写入权限。

– E 箭头：表示远程用户获得了非授权账号。

级别 2 和级别 3 包括诸如本地用户获取到了他们本不可以访问的文件的读写权限这类事件。当然，任何本地用户访问/tmp 目录都具有危险性，它能够潜在地铺设一条通向级别 3 的路。在级别 3，用户可以获取写访问权限（并由此过渡到级别 4 环境）。

级别 2 攻击是危险的，并很容易发展为级别 3、级别 4、级别 5 和级别 6。如果运行这种网络，请立即取得上述访问控制设备，如果不照此进行，某些人想破坏网络仅仅是时间问题。如果有可能，请监控所有流经端口 137~139 的消息，其间将产生共享进程。

本地攻击的难度不太大。所谓本地用户（Local user），我们认为是相对而言的。在网络世界中，本地用户是在本地网络的任一台机器上有口令，因而在某一驱动器上有一个目录的用户（无论那个目录的服务目的是什么）。

由本地用户启动的攻击几乎都是从远程登录开始的。对于 ISP，最好的办法是将所有 shell 账号放置于一个单独的机器上，也就是说，只在一台或多台分配有 shell 访问的机器上接受注册。这可以使日志管理、访问控制管理、释放协议和其他潜在的安全问题管理更容易些。还应该将存放用户 CGI 的系统区分离出来。这些机器应该隔离在特定的网络区段，也就是说，根据网络的配置情况，它们应该被路由器或网络交换机包围。其拓扑结构应该确保硬件地址不能超出这一特定区段。

针对这些利用访问控制营造所需环境的攻击，有两种涉及许可权的关键因素，每一种都能影响到级别 2 是否会升级到级别 3、4 或 5。这些因素是：

① 端口的错误配置。

② 软件中的漏洞。

第一种情况的发生是没有正确理解许可模式，不是每一个 UNIX 或 NT 系统管理员都是专家，经验是非常重要的。

第二种情况更加普遍，任何操作系统都有漏洞。对此问题尚未有直接的解决办法，因为大多数这种漏洞在软件加载时并不出现。唯一的办法是订阅每一种与故障、漏洞、系统密切相关的邮件列表。

级别 2 和级别 3 的主要攻击方法是社会管理邮件（电子邮件攻击的一种）：黑客会诱骗合法用户告知其机密信息或执行任务，有时黑客会假装为网络管理人员向用户发送邮件，要求用户提供系统升级的密码。

3．级别 4

– F 箭头：表示远程用户获得了特定文件的读权限。

级别 4 通常与外界能够访问内部文件相关。这种访问能做到的不只是核实特定文件是否存在，而且还能读这些文件。级别 4 还包含这样一些弱点，即远程用户无需账号就可以在服务器上执行有限数量的命令。由于服务器配置失误，有害 CGI 及溢出问题都可能引发这些漏洞大量出现。

密码攻击法是级别 4 中的主要攻击法，损坏密码是最常见的攻击方法。用户常常忽略他们的密码，密码政策很难得到实施。黑客有多种工具可以击破技术所保护的密码。一旦黑客拥有了用户的密码，他就拥有很多用户的特权。“密码猜想”是指手工敲入普通密码或通过编好的程序取得密码。一些用户选择简单的密码——如生日、纪念日或配偶名字，不遵循字母、数字混合的规则。对黑客来说要猜一串 6 个字生日数据不用花多长时间。最好的防卫方法便是严格控制进入特权。

4．级别 5、级别 6

– H 箭头：表示远程用户获得了特定文件的写权限。

– G 箭头：表示远程用户获得了根权限。

级别 5 和级别 6 产生于那些绝不应该发生的事被允许发生了的情况下。任何级别 5 和级别 6 的漏洞都是致命的。在这一阶段，远程用户可以读、写并执行文件（通常，他们综合各种技术来达到这一阶段）。级别 6 表示攻击者拥有这台机器的超级用户或管理员许可权。换句话说，攻击者具有对机器的全部控制权，可以在任何时刻完全关闭甚至毁灭此网络。

级别 5 和级别 6 的主要攻击是 TCP/IP 连续偷窃、被动通道听取和信息包拦截。这些都是为进入网络收集重要信息的方法，不像拒绝服务攻击，这些方法有更多类似偷窃的性质，比较隐蔽、不易被发现。

TCP/IP 连续偷窃指抓住连续数字，这些数字用来让黑客的信息包看起来合法化，当一个系统要求与其他系统对话，系统会交换 TCP 同时产生的数据，如果这些数据不是具有随意性的，黑客会收集这些数据的算法，被偷的突发事件会被用来把黑客伪装成一个或两个原始系统，允许他连接防火墙信息包的过滤器，这在连接 IP 时更有效。

一次成功的 TCP/IP 攻击能让黑客阻拦两个团体之间的交易，提供中间人袭击的良好机会，然后黑客会在不被受害者注意的情况下控制一方或双方的交易。

通过被动窃听，黑客会操纵和登记信息，也会从目标系统上所有可通过的通道找到可通过的致命要害。黑客会寻找联机和密码的结合点，认出申请合法的通道。

信息包拦截是指在目标系统中约束一个活跃的听者程序，以拦截和更改所有的或特别的信息的地址。信息可被改送到非法系统阅读，然后不加改变地送回给黑客。

1.3.6 反击措施

（1）级别 1 攻击的处理方法主要是：过滤进入地址并与攻击者的 ISP 联系。对防范拒绝服务攻击的方法感兴趣的读者请查看徐一丁先生的文章“分布式拒绝服务攻击（DDoS）原理及防范”，http://www-900.ibm.com/developerWorks/cn/security/se-ddos/index.shtml，此处不再赘述。

（2）级别 2 攻击可以在内部处理。根据 Gartner 调查表明，目前，70%的攻击仍然来自组织内部。基本做法就是冻结或清除攻击者的账号。级别 2 攻击者一般常伴随使用内部嗅探器，防范方法可以参考“防范网络嗅探”，http://www-900.ibm.com/developerWorks/cn/security/se-profromsniff/index.shtml。

（3）对级别 3、级别 4 和级别 5、级别 6 攻击的反应，如果经历过高于级别 2 的攻击，问题就严重了。

– 首先备份重要的企业关键数据。
– 隔离该网络网段，使攻击行为仅出现在一个小范围内。
– 允许行为继续进行。如有可能，不要急于把攻击者赶出系统，为下一步做准备。
– 记录所有行为，收集证据。

收集的证据包括系统登录文件、应用登录文件、AAA（Authentication, Authorization, Accounting 认证、授权、计费）登录文件，RADIUS（Remote Authentication Dial-In User Service）登录、网络单元登录（Network Element Logs）、防火墙登录、HIDS（Host-base IDS，基于主机的入侵检测系统）事件、NIDS（网络入侵检测系统）事件、磁盘驱动器、隐含文件等。

收集证据时要注意，在移动或拆卸任何设备之前都要拍照；在调查中要遵循两人法则，即在信息收集中要至少有两个人，以防止篡改信息；应记录所采取的所有步骤，以及对配置设置的任何改变，要把这些记录保存在安全的地方。

（4）进行各种尝试（使用网络的不同部分）以识别出攻击源。

（5）为了使用法律武器打击犯罪行为，必须保留证据。而形成证据需要时间，为了做到这一点，必须忍受攻击的冲击（虽然可以制定一些安全措施来确保攻击不损害网络）。对此情形，我们不但要采取一些法律手段，而且还要至少请一家权威的安全公司协助阻止这种犯罪。这类操作的最重要特点就是取得犯罪的证据，并查找犯罪者的地址，提供所拥有的日志。对于所搜集到的证据，应进行有效地保存。在开始时制作两份，一个用于评估证据，另一个用于法律验证。

1.4　开源软件网络安全概述

Linux 服务器运行的软件主要包括 Samba, VsFtp, OpenSSH, MySQL, PHP 和 Apache 等，这些软件大多数是开源软件，而且都在不停升级，稳定版和测试版交替出现。

什么是开源软件？字面意思是公开源代码的软件，它的英文为 Open Source Software，简称为 OSS。软件既然连源代码都公开，那么自然可以免费使用，一般概念上，它与自由软件（Free Software）是一个等价的概念，用户可以自由地对它进行研究、改进、传播，而无须付出任何费用。随着网络的流行，软件的安全性问题越来越成为人们讨论的焦点，每年大量的黑客攻击事件，病毒、蠕虫、木马的泛滥，使人们不得不高度重视软件的安全性问题。而与此同时，随着开源软件的应用越来越广泛，到底是开放源代码的软件更安全还是不开放源代码的商业软件（Commercial Software）更安全，成为长期以来人们一直关注的问题。

有人认为由于开源软件源代码是开放的，因此即使有漏洞和 Bug 也极易被人们发现并及时进行修改和完善，但是商业软件却由于不开放源代码，仅有少数人知道源代码，因此在漏洞和 Bug 方面，可能很晚才会发现，所以也极易造成破坏和损失。但也有人从另一个角度认为，由于开源软件的源代码开放，不法分子从中发现漏洞的机会将会更大，向其中加入病毒木马也更容易，而商业软件由于源代码保密，知悉源代码的人少，被发现漏洞的机会也会小一些。可以说双方都有自己的优点，也都有自己的缺点。

那么，究竟谁会更安全一些呢？事实证明，很多软件使用者，还有一些研究机构，从他们的使用和研究状况来看，开源软件的安全性确实要好一些，而这很大程度上就在于开源软件的开放性和随时弥补性。

事实上，不管是开放源代码的软件还是不开放源代码的软件，漏洞都无可避免，可以毫不夸张地说，所有的软件都有漏洞，只是这些漏洞是不是能及时被发现并进行弥补。软件安全与否，与是否开

放源代码关系并不大，把源代码公开并不一定能确保代码本身的安全性，同样，封闭源代码也不一定使代码本身变得不安全。从某种程度上讲，人们所依赖的所谓软件的“安全性”其实更多的是一种臆想和希望，而不是现实。即使是像微软这样的软件巨人，也以每年要在自己的软件上修正多少漏洞，打上多少补丁而著名，而这些还仅仅是已经发现的漏洞，更不用说还没有被发现的漏洞。我们要知道“没有绝对安全的软件”，这是一个基本道理。

因此，追求软件代码自身的安全性，做到让软件本身没有任何漏洞，这几乎是不可能的，人们所谓的软件安全性，更大程度上在于对漏洞的及时发现以及修补。而开源软件，恰恰在这方面具有商业软件所不具有的先天优势。现在的开源软件，很大程度上是一种全球的智慧，是全体软件设计研究人员共同的兴趣和爱好的结晶，其中积累了大部分人的智慧。由于其开放性和共享性，它会充分受到广大的软件研究者和使用者的评审，因而漏洞会较少。即使有漏洞，也能被及时发现，出现问题之后也能更快更容易地进行弥补。所以，从这个角度上来说，开源软件的确会比商业软件更安全。

Linux 确实有自己的安全弱点。最常见的弱点是对于某些高级技术缺乏可靠的本地支持。厂商一般开发的硬件和相关的驱动程序软件只为大多数 Windows 用户使用。Linux 团体通常对这些产品做逆向工程处理，使这些产品兼容开源软件操作系统。这首先就使他们的工作没有预见性。在某些情况下，可兼容的 Linux 硬件要比 Windows 落后几个月甚至几年。幸运的是，由于 IBM 和 Novell 支持开源软件标准，帮助优化兼容过程，这个问题并没有引起多大麻烦。

在 Linux 的图形界面接口之外，Linux 的命令行是非常复杂的，通常是不容易学会的。这就延缓了管理员掌握加强系统安全的时间。Linux 主要用做支持网络功能的操作系统，默认安装时不必要地启动了很多网络应用程序。这就可能造成不为人知的安全漏洞。幸运的是，让管理员操作简单的命令行工具弥补了这些弱点。

最好是了解 Linux 和 Windows 这两种操作系统相对的优点，在充分分析各个系统的弱点的同时，根据业务的主要需求来选择操作系统。

评定安全等级的更客观的方法是跟踪一个特定的套装软件发布的修复漏洞的补丁数量。当与 Linux 进行对比的时候，这种衡量方法表明 Windows 似乎安全漏洞更多。美国计算机应急反应小组最近发表的安全漏洞测评报告称，微软的 Windows 出现了 250 次安全漏洞，其中有 39 个安全漏洞的危险程度达到了 40 分或者 40 分以上。而 Red Hat Linux 只有 46 次安全漏洞，其中只有 3 个安全漏洞的危险程度在 40 分以上。对于这两个操作系统的对比已经有数千份报告了，但是，像这种独立的政府机构发布的报告是最值得考虑的。

在安全方面存在这种差别是有充分理由的。例如，Linux 的开源软件开发方式有助于更容易地暴露错误，这是微软不具备的优势。微软的 Windows 另一个不利因素是其许多应用程序依靠远程程序调用。远程程序调用是计算机内部通信的一种方式，无法预知地和主动地分配通信通路。与限制使用远程程序调用的 Linux 相比，这种方式将使 Windows 的防火墙没有 Linux 那样严格。

有些安全差别不仅系统管理员可以看到，最终用户也可以看到。例如，Windows 受到的病毒感染最多，促使大多数用户购买杀毒软件以保证自己的系统安全。最近，流氓软件和间谍软件开始入侵 Windows 系统，在用户浏览网络时不明智地下载和启动流氓软件、间谍软件之后，流氓软件、间谍软件就会暗地里获取和发布用户的个人信息并且骚扰用户。使用管理员权限和普通的用户账号都可以操作 Windows 和 Linux 系统。但是，某些第三方 Windows 应用软件没有严格坚持这个特点，经常需要管理员的权限才能正确运行软件。因此，这些用户发起的病毒攻击的破坏性是很大的。Linux 应用软件通常都遵守这个安全要求，因此很少被攻击者利用。Windows 易学易用的目的达到了，但是，其代价是牺牲了全面的安全。此外，Windows 需要兼容不安全的老版本的软件，这个缺点是 Linux 所没有的。

1.5　本章小结

操作系统是计算机系统的灵魂，维护着系统的底层，对内存、进程等子系统进行管理和调度。如果操作系统本身出现了漏洞，其影响将会是致命的。操作系统的内核，对于网络安全是至关重要的。目前，内核的维护主要分两种模式：对于私有操作系统，如 Windows/Solaris 等，由于个人用户不能直接接触其源代码，其代码由公司内部开发人员维护，其安全性由同样的团队保证，内核的修正与其他应用程序一样，以 patch/SP 包的方式发布；对于 Linux 这样的开放式系统，从机制上讲，全世界的开发人员都能获得源代码，从而找出其中的纰漏，但是同时，如果网络管理人员不能及时更新内核，也会留下安全隐患。影响操作系统安全的因素有很多，从编程水平到用户的使用水平等，都将影响到系统的安全。仅仅通过开放或者封闭源代码，都不能从根本上解决安全问题。如果你是一个 Linux 网管员，你经常需要上相应的网站查看是否有补丁，是否有了 bug fix，是否需要升级。千万不要报侥幸心理，否则一个 shell 脚本就可能拿下你的网站。套用一句名言“你的服务器永远可能在第二天被黑客接管”。