大数据、云计算、人工智能、信息安全人才培养丛书
“互联网+”新形态一体化精品教材

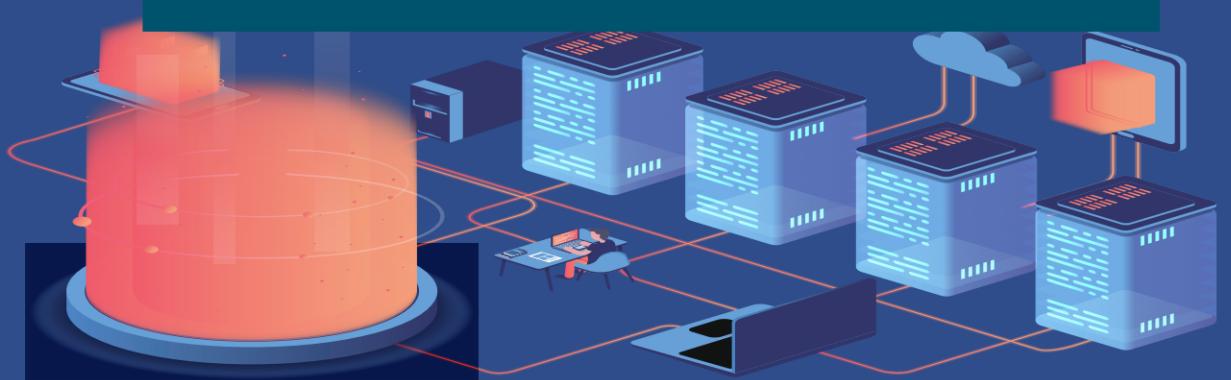
Linux服务器安全高级运维

Linux FUWUQI ANQUAN GAOJI YUNWEI

主编◎陈　登　赵　滨　李　翱

扫一扫
学习资源库

◆微课视频
◆教学课件
◆电子教案



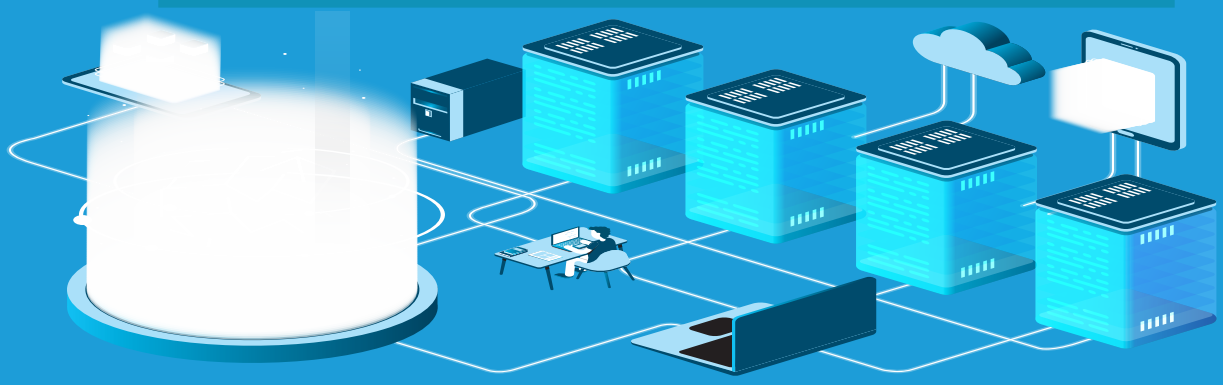
上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

大数据、云计算、人工智能、信息安全人才培养丛书
“互联网+”新形态一体化精品教材

Linux服务器安全高级运维

Linux FUWUQI ANQUAN GAOJI YUNWEI

主　编　陈　登　赵　滨　李　翱
副主编　马晓涛　杨　晶　张　莹



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

Linux 是全球流行的服务器平台，本书从 Linux 系统安全加固、运维故障排查、自动化运维、服务器集群架构等方面讲解构建大规模和高性能 Linux 服务器集群所需的技术、工具、方法和技巧及安全配置，由浅入深，全面系统地与实践紧密结合。所有理论知识、方法、技巧和案例都来自实际环境，涉及面广，从基本操作、服务器安全加固、基本网络应用到高级网络服务器运维集群和自动化运维知识体系，方便读者结合自己的实际工作环境灵活应用。

本书适合作为高校计算机相关专业，特别是网络空间安全、信息安全管理、网络工程等专业有关课程的教学用书，也适合 Linux 初学者、爱好者、程序员或者从事网络安全管理、系统架构、安全运维等专业技术人员使用。

图书在版编目（CIP）数据

Linux 服务器安全高级运维 / 陈登，赵滨，李翱主编
. — 上海：上海交通大学出版社，2021（2022 重印）
ISBN 978-7-313-24531-1

Ⅰ. ① L… Ⅱ. ①陈… ②赵… ③李… Ⅲ. ① Linux 操作系统 Ⅳ. ① TP316.85

中国版本图书馆 CIP 数据核字（2021）第 020090 号

Linux 服务器安全高级运维
Linux FUWUQI ANQUAN GAOJI YUNWEI

主　　编：陈登　赵滨　李翱
出版发行：上海交通大学出版社
邮政编码：200030
印　　制：北京华创印务有限公司
开　　本：889 mm × 1194 mm　1/16
字　　数：336 千字
版　　次：2021 年 1 月第 1 版
书　　号：ISBN 978-7-313-24531-1
定　　价：56.00 元

地　　址：上海市番禺路 951 号
电　　话：6407 1208
经　　销：全国新华书店
印　　张：15
印　　次：2022 年 7 月第 2 次印刷

前言

本书以 Linux 服务器安全加固、安全运维、高可用集群为核心内容，多方面深入讲解了如何构建高性能、高可用、安全、稳定的 Linux 服务器。书中观点得到奇虎 360、安恒、蓝盾、永信至诚、易霖博等国内知名网络安全公司研究机构的充分认可。内容由浅入深，适合作为高校计算机相关专业，特别是网络空间安全、信息安全管理、网络工程等专业有关课程的教学用书。也可以作为 Linux 爱好者、程序员或者从事网络安全管理、系统架构、安全运维等专业技术人员的参考用书。

本书深入阐述了 Linux 服务器的安全运维，包括初识 Linux 服务器安全运维，Linux 网络安全运维，数据安全工具 DRBD、extundelete，Linux 系统运维故障排查思路，Linux 故障排查案例实战，轻量运维利器 pssh、pdsh 和 mussh，分布式监控系统 Ganglia，基于 Nagios 的分布式监控报警平台 Centreon，构建智能化监控报警平台等 9 个项目。

本书在编写上具有如下特点：

（1）采用任务驱动、案例引导的写作方式，从工作过程出发，从项目出发，以实际应用为主线，突破以知识点的层次递进为理论体系的传统模式，将职业工作过程系统化，按照工作过程来组织和讲解知识，培养学生的职业技能和职业素养。

（2）根据读者的学习特点，考虑到因学生基础参差不齐而给教师授课带来的困扰，在编写过程中将内容划分为多个任务，每一个任务又划分了多个子任务，以“做”为中心，“教”和“学”都围绕着“做”展开，在学中做，在做中学，从而提高学生的自我学习能力。

（3）紧跟行业技能发展，着重于当前主流技术讲解，与行业联系密切，从而达到学以致用的目的。

此外，本书作者还为广大一线教师提供了服务于本书的教学资源库，有需要者可致电 13810412048 或发邮件至 2393867076@qq.com。

本书由多位安全运维与网络空间领域的资深任课教师共同编写，由于编写时间仓促，加之网络技术发展迅猛，书中存在的不足和疏漏之处，敬请广大读者批评指正，以便再版时修订完善，在此表示衷心的感谢。

目录

项目 1　初识 Linux 服务器安全运维 / 1

项目 2　Linux 网络安全运维 / 29

项目 3　数据安全工具 DRBD、extundelete / 51

项目 4　Linux 系统运维故障排查思路 / 81

项目 1 初识 Linux 服务器安全运维

项目目标

1. 了解用户与文件系统概念。
2. 掌握文件权限与分配设置。
3. 了解系统软件安装工具。
4. 了解后门程序及检测工具使用。
5. 了解服务器遭受攻击后的分析及处理过程。

知识导图

笔记

项目导入

Linux 系统之所以会成为目前最受关注的系统之一，主要因为它是免费的，并且系统具有开放性，可以随时取得程序的源代码，这对于程序开发人员十分重要。作为非常优秀的操作系统平台，它的优势体现在以下几个方面：

（1）Linux 开源，可以减少成本，可定制性也更高。

（2）由于 Linux 倡导简单即是美，内核没有提供过多的功能，许多功能是由其他软件提供的，所以可以关闭大多数与任务不相关的程序（如 gnome）；有些地方也没有 Windows “智能”（如权限控制不会综合考虑），所以相对于 Windows 运行效率较高。

（3）同样由于 Linux 倡导简单，所以相对于 Windows 漏洞更少，如果管理者注意按时检查、维护，被入侵的概率是比较小的。

（4）Linux 主要使用命令行控制，所以在命令行方面发展出了非常方便的特性，有许多工具性质的程序可以使用。而命令行在批处理、自动化方面有着先天的优势，这是图形界面难以具备的。

（5）由于上面的原因，许多专业性的程序只能在 Linux 上运行，或者主要在 Linux 上运行，而在 Windows 上运行时功能不全，容易崩溃。比如 Hadoop、octave 等。这些工具导致更多的公司和个人使用 Linux，并且在 Linux 上开发更多相应的工具。

正是因为 Linux 具备了开源、灵活、性能安全稳定这些优点，在全球范围内的很多行业和企业选择把 Linux 作为其服务器平台。同时，Linux 服务器也担负起了保证业务运转与行业数据安全的重大使命。

安全总是相对的，再安全的服务器也有可能遭受攻击。如何保证 Linux 服务器的正常稳定运行，就涉及安全运维的问题。运维安全是企业安全保障的基石，不同于 Web 安全、移动安全或者业务安全，运维安全环节出现问题往往会比较严重。

负责运维的专业人员需要有较高超的专业技术，概括起来主要应掌握以下几方面的专业知识与应用：

（1）要求熟练使用 Linux 常用的命令。

（2）网络配置及华为思科的网络设备使用与配置。

（3）要懂性能调优。包括 lamp 或者 lnmp 的性能调优，也包括 Linux 操作系统调优。

（4）要懂数据库 MySQL 或者 NoSQL（如 MongoDB）。

（5）要懂编程语言，最基本的首推 Shell，还要学习 Perl、Python，甚至 Ruby 和 C++ 等，还得熟练掌握 awk、sed、grep 以及正则表达式。

（6）要懂一些调试排错工具的使用，如 htop、dstat、strace、systemtap、iostat、sar 等。

（7）企业实体应建立高效的预警响应机制。在运维安全提醒建设到相对完善的情况下，企业是相对安全的。但在有新漏洞或遭受攻击的情况下，唯一能保障的就是在中招后提示响应和修复的速度。

（8）养成并奉行良好的工作习惯。运维人员要有始终如一的安全意识，比如对用户及口令的严格设置与强化、严防后门程序泄露的风险以及及时检测调节系统性能，做好工作日志等。

任务 1.1　账户和登录安全

笔记

子任务 1.1.1　删除特殊的用户和用户组

Linux 系统初次安装完成后默认提供了各种类型的账号，这些信息以记录的形式被保存在 /etc/passwd 文件中。其中有一大批以 /sbin/nologin 结束的账号并不能用来登录 Linux 系统，而是一些保持进程或服务正常运行的系统账号。为了安全起见，不需要的用户或用户组是可以删除的。如图 1-1 所示。

```
[root@localhost ~]# cat /etc/passwd
```

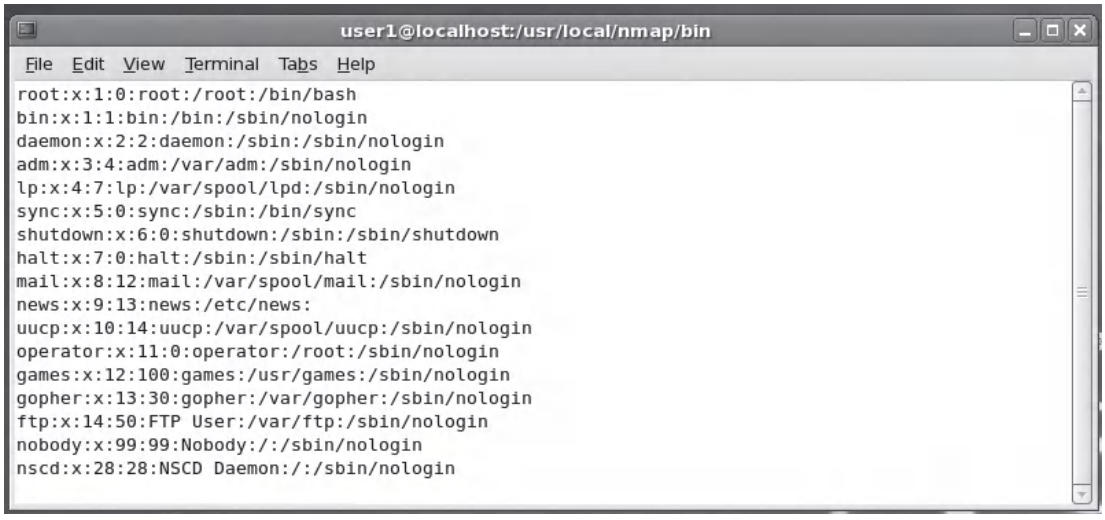


图 1-1　用户管理文件 /etc/passwd

删除用户语法格式：

userdel　用户名

例如：

```
[root@localhost ~]#  userdel  games
```

删除用户组语法格式：

groupdel　组名

例如：

```
[root@localhost ~]# groupdel  games
```

子任务 1.1.2　关闭系统不需要的服务

Linux 是一种可以设置运行级别的操作系统，级别不同开机自动启动的服务也会有所不同。但是，在众多的服务当中，有些需要设置才能起作用，而有些针对用户的需求来说是根本没必要启动的。开启的服务越多所花费的时间越长，因此关闭部分服务可以加快开机启动速度。

笔记

语法格式：

chkconfig [--add][--del][--list][系统服务名]

chkconfig [--level < 等级代号 >][系统服务名][on/off/reset]

例如：

```
[root@local ~]# cd  /etc/rc.d/init.d
// 了解不同运行级别下的服务
// 显示全部服务开启 / 关闭列表
[root@local ~]# chkconfig  --list
// 关闭运行级别 3 和 5 下的防火墙
[root@local ~]# chkconfig --level 35 iptables off
// 开启运行级别 3 和 5 下的防火墙
[root@local ~]# chkconfig --level 35 iptables on
```

子任务 1.1.3 设置密码安全策略

运维人员相当重要的一个工作就是管理系统账号，保证系统账号安全登录有很多种方法。

1. 密码认证

1）限制登录次数防止攻击

```
[root@localhost ~]#vi /etc/pam.d/system-auth
// 添加下列信息
auth  required  pam_tally.so  deny=5
// 设置连续登录超过 5 次将被锁定
account required  pam_tally.so
```

2）设置复杂密码保证安全

```
[root@localhost ~]# vi /etc/login.defs
  PASS_MAX_DAYS 90      // 设置口令失效天数为 90 天
  PASS_MIN_DAYS 0       // 本次修改与下次修改之间的天数间隔
  PASS_MIN_LEN  6       // 指定密码设置的最小长度
  PASS_WARN_AGE 7       // 指定密码失效前多少天发送提醒信息
```

密码设置应尽可能复杂，包含数字、字母、下划线、特殊符号等，能起到一定的防护作用，但同时也会面临密码丢失的风险。除了密码认证之外，还有一种更安全的认证方式——密钥认证。

2. 密钥认证

密钥认证是一种新型认证方式，公用密钥存储在远程服务器上，专用密钥保存在本地，当需要登录系统时，通过本地专用密钥和远程服务器的公用密钥进行配对认证，认证成功，就可以实现系统登录。这种认证方式避免了被暴力破解的危险，同时只要保存在本

地的专用密钥不被黑客盗用，攻击者一般无法通过密钥认证的方式进入系统。因此，在 Linux 下推荐用密钥认证方式登录系统。

笔记

常见远程维护管理客户端工具包括 SecureCRT、putty、Xshell 等，Linux 系统提供 SSH（Secure Shell）服务，两部分共同合作才能完成密钥加密。

1）SecureCRT 部分配置

（1）远程计算机下载并安装 SecureCRT 软件后，运行该软件，单击菜单栏“工具”后选择“创建公钥”。

（2）弹出“密钥生成向导”对话框，阅读提示信息后，单击“下一步”。

（3）单击“密钥类型”下拉箭头，选择 RSA 方式，单击“下一步”。

（4）在“通行短语”文本框中输入自定义口令，再次输入相同口令到“确认通行短语”文本框中，在“注释”文本框中输入自定义文本提示信息，单击“下一步”。

（5）在“密钥长度（位）”文本框中输入推荐值 1024，单击“下一步”。

（6）系统开始生成密钥。

（7）为生成的密钥选择或自定义一个文件名和存放目录，单击“完成”。

2）Linux 部分配置

（1）上传公钥文件到 Linux 系统，通过 Linux 命令将公钥文件导入 SSH 服务。

```
[zhy@localhost ~]$ mkdir /home/zhy/.ssh  // 测试用户 zhy，为 zhy 用户创建宿主目录下的隐含文件
[zhy@localhost ~]$ chmod 700 /home/zhy/.ssh // 设置该隐含文件的安全访问权限为 700
```

（2）SecureCRT 先选择非密钥认证方式，将 SecureCRT 配置中第 7 步生成的文件 Identity.pub（公钥）上传到 Linux 服务器测试用户 zhy 刚刚创建好的隐含目录 .ssh 下，然后执行更名操作。

```
[zhy@localhost~]$ ssh-keygen -i -f Identity.pub >> /home/zhy/.ssh/authorized_keys
```

（3）在 SecureCRT 客户端软件上“新建”一个 SSH2 连接。在“快速连接”窗口“协议”下拉列表中选择“SSH2”，在“主机名”文本框中输入 Linux 服务器 IP 地址“172.22.55.21”，“端口”默认“22”，“防火墙”默认“None”，在“用户名”文本框中输入“zhy”，在“鉴权”一栏中选择“公钥”方式，然后单击右边的“属性”按钮。

（4）弹出“公钥属性”窗口，选择“使用身份或证书文件”，单击“...”按钮找到 SecureCRT7 中生成的公钥文件 Identity.pub，单击“确定”按钮。

（5）修改服务器端 SSH 配置文件，设置 PublicKey 认证方式验证用户。

```
[root@localhost ~]# vi          /etc/ssh/sshd_config
Protocol 2                      // 仅允许使用 SSH2
PubkeyAuthentication yes        // 启用 PublicKey 认证
AuthorizedKeysFile .ssh/authorized_keys #PublicKey 文件路径
PasswordAuthentication no       // 不使用口令认证
```

笔记

最后重启 sshd 服务，执行如下命令：

```
[root@localhost ~]# /etc/rc.d/init.d/sshd restart
```

等 sshd 服务启动完毕，就可以利用 SecureCRT 通过 PublicKey 认证远程登录 Linux 系统了。

子任务 1.1.4 合理使用 su、sudo 命令

为了保证服务器的安全，Linux 系统默认只允许以普通用户身份登录图形化界面。需要执行超级权限工作时可以通过 su（switch user）命令转换身份来实现。

语法：su [欲切换账户名]

例如：

```
[user1@localhost ~]$ su  // 由普通用户切换到管理员不需要输入用户名
输入管理员口令：123456
[root@localhost ~]#
[root@localhost ~]# su  user1 // 由管理员切换到 user1 不需要输入口令
[user1@localhost ~]$
```

su 命令是一个快速简洁实现用户身份切换的工具，但这种方式无法保证服务器的访问安全。因此，Linux 系统还提供了另外一种方式——sudo。sudo 命令无需转换操作者身份，可以实现不泄露管理员密码给普通用户，只是通过管理员身份修改指定命令的权限来实现普通用户对系统的管理。

```
[root@localhost ~]# visudo  -f  /etc/sudoers
// 管理员编辑 sudo 主配置文件
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##  user  MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root  ALL=(ALL)  ALL
// 第一个 ALL：指定用户可以执行特殊命令的所有网络主机的主机名；
// 第二个（ALL）：指定执行特殊命令时身份的用户名；
// 第三个 ALL：指定有执行权限的命令名列表；
```

```
    user1  ALL=(ALL)   ALL
    // 允许 user1 在任何一台主机上以任何身份执行所有命令
    user2  ALL=(ALL)   /sbin/ifconfig
    // 允许 user2 在任何主机上执行命令 ifconfig
    user3  localhost=   /sbin/ifconfig
    // 允许 user3 在本机上执行 ifconfig
    user4  localhost=NOPASSWD:/sbin/ifconfig
    // 允许 user4 不输入密码使用 ifconfig 单击“ESC”，输入“：wq”，保存并退出主配置文件 sudoers。
    [usesr1@localhost ~]$ sudo ifconfig
    输入管理员口令：
    [user4@localhost ~]$ sudo   ifconfig   // 不需要口令
```

笔记

子任务 1.1.5　删减系统登录欢迎信息

登录 Linux 系统前，总会显示几行欢迎信息或版本信息，这些信息虽然可以为管理者带来便利，但是也容易被黑客利用。因此编辑信息显示配置文件，以实现安全访问。

通过本地终端、本地虚拟控制台登录，显示的信息存放于 /etc/issue 文件中；

通过远程 SSH 或 telnet 登录，显示信息存放于 /etc/issue.net 文件中。

```
[root@localhost ~]# cat   /etc/issue   // 该文件允许使用转义符
[root@localhost ~]# cat   /etc/issue.net   // 该文件不允许使用转义符
[root@localhost ~]# cat   /etc/redhat-release
// 查看 Redhat、CentOS 的版本信息
[root@localhost ~]# cat   /etc/motd   // 登录成功后显示管理员提示信息
```

子任务 1.1.6　禁止 Control-Alt-Delete 键盘关闭命令

Linux 系统默认情况下允许任何人按下“Control+Alt+Delete”组合键自动重启，这是很不安全的。因此为了系统安全应禁用“Control+Alt+Delete”组合键重启系统功能。

```
[root@localhost ~]# vi /etc/inittab   # centos 版本下的配置文件
```

Centos6.x 及以下会找到如下这行：

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

在这行内容之前加上“#”注释，然后执行：

```
[root@localhost ~]# init q   // 重新加载（reload）配置文件
```

笔记

在 CentOS7 以上版本中会找到如下内容：

```
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

在这行之前加上“//”注释掉即可。

任务 1.2 远程访问和认证安全

子任务 1.2.1 远程登录取消 telnet 而采用 SSH 方式

telnet 是早期 Internet 远程登录服务的标准协议和主要方式。终端使用者计算机用它成功连接到服务器后，输入各种命令，以不加密的形式传递给服务器并运行，就像直接在服务器的控制台上输入一样，因此非常不安全。目前取而代之的是 SSH 服务。SSH 也是由客户端和服务器端软件组成的，客户端可以使用的软件有 SecureCRT、putty、Xshell 等，服务器端运行的是一个 sshd 服务。SSH 不但可以加密压缩所有传输数据，还能够防止 DNS 和 IP 欺骗。

```
[root@localhost ~]# vi  /etc/ssh/sshd_config // 打开主配置文件
```

主配置文件中部分配置选项的含义如下：

#Port 22 ：设置 sshd 监听的端口，建议更改默认的 22 端口，选择陌生数字作为端口号。

#Protocol 2，1 ：设置使用的 SSH 协议的版本为 SSH1 或 SSH2，SSH1 版本有缺陷和漏洞。

#Protocol 2：选择 Protocol 2。

#ListenAddress 0.0.0.0：设置 sshd 服务器绑定的 IP 地址。

#HostKey /etc/ssh/ssh_host_dsa_key：设置服务器密匙文件的路径。

#KeyRegenerationInterval 1h ：设置两次密钥间隔多长时间系统自动重新生成服务器的密钥。

#ServerKeyBits 768：用来定义服务器密钥长度的字节数。

#SyslogFacility AUTHPRIV ：设置记录来自 sshd 消息的时刻，是否给出“ facility code”。

#LogLevel INFO：用来设置 sshd 日志消息的级别。

#LoginGraceTime 2m：设置用户登录失败切断连接前服务器需要等待的时间。

#PermitRootLogin yes ：设置超级用户 root 能否用 SSH 登录，远程登录比较危险，建议设置为“no”。

#StrictModes yes ：设置 SSH 在接收登录请求之前是否检查用户根目录和 rhosts 文件的权限和所有权。

#RSAAuthentication yes ：设置是否开启 RSA 密钥验证，只针对 SSH1，采用 RSA 密钥登录方式时，开启此选项。

#PubkeyAuthentication yes：设置是否开启公钥验证，采用公钥验证方式登录时，开启此选项。

#AuthorizedKeysFile .ssh/authorized_keys：设置公钥验证文件的路径，与 PubkeyAuthentication 配合使用。

#IgnoreUserKnownHosts no：设置 SSH 在进行 RhostsRSAAuthentication 安全验证时是否忽略用户的“$HOME/.ssh/known_hosts”文件。

#IgnoreRhosts yes：设置验证的时候是否使用“~/.rhosts”和“~/.shosts”文件。

#PasswordAuthentication yes：设置是否开启密码验证机制，使用密码登录系统时设置为“yes”。

#PermitEmptyPasswords no：设置是否允许用口令为空的账号登录系统，必须选择“no”。

#ChallengeResponseAuthentication no：禁用 s/key 密码。

#UsePAM yes：不通过 PAM 验证。

#X11Forwarding yes：设置是否允许 X11 转发。

#PrintMotd yes：设置 sshd 是否在用户登录的时候显示“/etc/motd”中的信息，可以在 /etc/motd 中加入警告信息，以震慑攻击者。

#PrintLastLog no：是否显示上次登录信息，设置为“no”表示不显示。

#TCPKeepAlive yes：选择“yes”防止死连接。

#UseDNS no：是否使用 DNS 反向解析，这里选择“no”。

#MaxStartups 10：设置同时允许几个尚未登入的联机。

子任务 1.2.2 合理使用 shell 历史命令记录功能

Linux 下通过 shell 终端输入的命令都会被记录下来。

```
[root@localhost ~]# history  // 提供 history 命令查看历史记录
[root@localhost ~]# vi  ~/.bash_history  // 每个用户宿主目录下都有一份属于自己的 .bash_history
```

为了方便系统运维人员的使用，Linux 默认在 .bash_history 文件中保存了 500 条使用过的命令。但是，这种策略实际是非常危险的。一方面，历史记录可能会泄露管理员的各种操作记录，因此需要限制历史记录的条数或者每次操作后删除历史记录文件。另一方面，管理员需要通过历史记录查询黑客登录服务器所执行过的历史命令操作。因此，只有合理保护或备份 .bash_history 文件才是最重要的。

```
[root@localhost ~]# vi /etc/profile
// 编辑系统配置文件设置历史记录条数及格式
HISTFILESIZE=30  // 设置 .bash_history 文件中历史记录总数
HISTSIZE=30  // 设置 history 命令输出的记录条数
#history
USER_IP=`who -u am i 2>/dev/null| awk '{print $NF}'|sed -e 's/[()]//g'`  HISTDIR=/usr/share/.history
```

笔记

```
if [ -z $USER_IP ]
then
USER_IP=`hostname`
fi
if [ ! -d $HISTDIR ]
then
mkdir -p $HISTDIR
chmod 777 $HISTDIR
fi
if [ ! -d $HISTDIR/${LOGNAME} ]
then
mkdir -p $HISTDIR/${LOGNAME}
chmod 300 $HISTDIR/${LOGNAME}
fi
DT=`date +%Y%m%d_%H%M%S`
export HISTFILE="$HISTDIR/${LOGNAME}/${USER_IP}.history.$DT"
export HISTTIMEFORMAT="[%Y.%m.%d %H:%M:%S]"
chmod 600 $HISTDIR/${LOGNAME}/*.history* 2>/dev/null
```

这段代码将每个用户的 shell 命令执行历史以文件的形式保存在 /usr/share/.history 目录中，每个用户一个文件夹，并且文件夹下的每个文件以 IP 地址加 shell 命令操作时间的格式命名。

子任务 1.2.3 启用 tcp_wrappers 防火墙

Linux 系统的安全除了来自于严格的权限控制外，防火墙也是保证安全的一个重要手段。Linux 系统提供了两层防护措施。第一层是通过 IP 信息包过滤机制的 iptables，iptables 防火墙通过监控系统的运行状况，阻拦网络中的一些恶意攻击，接下来的第二层就是 tcp 层存取控制过滤机制的 tcp_wrappers 了。通过 tcp_wrappers 可以实现对系统中某些服务的开放与关闭，从而更有效地保证系统安全运行。

tcp_wrapper 提供了 tcpd 守护进程和 libwrap.so 开发库，通过 tcpd 进程访问 /etc/hosts.allow（允许访问的规则文件）和 /etc/hosts.deny（拒绝访问的规则文件），以及服务是否依赖 libwrap.so 库文件，来检验系统服务是否支持 tcp_wrapperd 的集中验证功能。

```
[root@localhost ~]# rpm -qa|grep tcp  // 检测 tcp_wrapper 是否安装
[root@localhost ~]# ldd `which sshd`
// 通过检测 sshd 的依赖关系来验证是否可以使用集中验证功能
[root@localhost ~]# vi  /etc/hosts.allow
// 查看 hosts.allow 规则，确定允许访问
service:host(s) [:action]
```

主要参数含义如下：

service：被集中监控的服务名，例如 sshd、vsftpd、sendmail 等。

host(s)：允许或拒绝访问的主机名或者 IP 地址，例如 192.168.1.0，www.apache.org，192.168.1.。

action：动作，符合条件后所采取的动作。

ALL:ALL EXCEPT 192.168.1.12　// 除 192.168.1.12 之外任何主机允许使用任何服务访问服务器

sshd: 172.22.91.56　// 允许 172.22.91.56 使用 sshd 远程登录服务器

一般情况下，Linux 会首先查看 /etc/hosts.allow 文件，如果远程登录的计算机满足该文件设定，则不再查看 /etc/hosts.deny 文件；相反，未在 hosts.allow 文件中设定，就会继续查看 /etc/hosts.deny 文件，如果满足 hosts.deny 的规则，此主机被限制为不可访问 Linux 服务器。如果两个文件中都没有设定，此主机默认是可以访问 Linux 服务器的。因此，在设定好 /etc/hosts.allow 文件访问规则之后，只需设置 /etc/hosts.deny 内容为“所有计算机都不能登录状态”即可。

```
[root@localhost ~]# vi  /etc/hosts.deny  // 设置不允许登录的规则
sshd:ALL  // 禁止所有计算机使用 ssh 远程登录服务器
```

这样，一个简单的 tcp_wrappers 防火墙就设置完成了。

任务 1.3　文件系统安全

子任务 1.3.1 锁定系统重要文件

Linux 中有很多非常重要的系统文件，为了防止 root 用户因意外操作而影响整个系统的正常运行，经常会针对文件进行锁定操作以保证安全。锁定文件操作主要是通过改变文件或目录属性来实现的。

更改文件或目录属性语法格式：

chattr [-RVf] [-v version] [mode] 文件或目录 ...

主要参数含义如下：

-R：递归修改所有的文件及子目录。

-V：详细显示修改内容，并打印输出。

其中 mode 部分用来控制文件的属性如表 1-1 所示。

表 1-1　mode 部分属性

属　性	说　明
+	在原有参数设定基础上，追加参数
–	在原有参数设定基础上，移除参数
=	更新为指定参数设定
A	文件或目录的 atime (access time) 不可被修改（modified），可以有效预防例如手提电脑磁盘 I/O 错误的发生

笔记

笔记

续表

属　性	说　明
S	硬盘 I/O 同步选项，功能类似 sync
a	即 append，设定该参数后，只能向文件中添加数据，而不能删除，多用于服务器日志文件安全，只有 root 才能设定这个属性
c	即 compresse，设定文件是否经压缩后再存储。读取时需要经过自动解压操作
d	即 no dump，设定文件不能成为 dump 程序的备份目标
i	设定文件不能被删除、改名、设定链接关系，同时不能写入或新增内容。i 参数对于文件系统的安全设置有很大帮助
s	保密性地删除文件或目录，即硬盘空间被全部收回
u	与 s 相反，当设定为 u 时，数据内容其实还存在磁盘中，可以用于 undeletion

例如：

```
[root@localhost ~]# chattr +i /etc/hosts
// 给文件 hosts 设置 i 属性锁定，防止修改和删除
[root@localhost ~]# vi /etc/hosts   // 提示警告信息
[root@localhost ~]# rm  /etc/hosts   // 提示警告信息
[root@localhost ~]# chattr +a /etc/profile
// 给文件 profile 设置 a 属性进行锁定，防止删除
[root@localhost ~]# vi /etc/profile   // 允许编辑操作
[root@localhost ~]# rm /etc/hosts   // 不允许删除操作
```

对重要的文件进行加锁后，如果涉及软件的升级等操作则需要先去掉有关目录或文件的 i、a 属性，否则将无法进行安装或升级。另外，chattr 命令不能保护 /、/dev、/tmp、/var 等目录。因为，如果根目录有不可修改的属性，那么系统根本无法工作。/tmp 是个临时目录，会有很多应用程序和系统程序需要在这个目录下建立临时文件；/var 是系统和程序的日志目录，如果设置为不可修改属性，那么系统将无法写入日志。

子任务 1.3.2　文件权限检查和修改

严格的文件权限控制是 Linux 安全访问的一个主要手段，不正确的权限设置将直接威胁系统的安全，因此运维人员要及时检测并发现问题权限设置，防患于未然。

语法格式：find　pathname -options [-print -exec -ok ...]

主要参数含义如下：

pathname：指定查找目标的路径范围。

-print：find 命令将匹配的文件输出到标准输出。

-exec ：find 命令对匹配的文件执行该参数所给出的 shell 命令，相应命令的形式为“'command' {} \;”。

-ok：与 -exec 的作用相同，且更为安全，在执行每一条命令之前，都给出提示让用户

来确定是否执行。

-options：部分选项。

-name：按照文件名查找文件。

-perm：按照文件权限查找文件，权限位不足 3 位以 0 补齐。

- find -perm mode ：表示严格匹配，即文件权限位转换成对应的十进制数字与 mode 一模一样。
- find -perm -mode：表示 mode 中转换成二进制的 1 在文件权限位里面必须匹配。
- find -perm +mode ：与 -mode 的区别是 +mode 只需其中的任意一个 1 的部分被匹配。

-nogroup：查找无有效所属组的文件，即该文件所属的组在 /etc/groups 中不存在。

-nouser：查找无有效属主的文件，即该文件的属主在 /etc/passwd 中不存在。

-type：查找某一类型的文件，具体如下：

- b -：块设备文件。
- d -：目录。
- c -：字符设备文件。
- p -：管道文件。
- l -：符号链接文件。
- f -：普通文件。

例如：

```
// 查找 / 目录下权限为 777 的文件
[root@localhost ~]# find / -type f -perm 777
// 权限为 777 的文件允许任何人以任何方式运行，如果是一些重要命令文件就相当于直接暴露给了所有用户，系统比较危险，检查到此类文件后应及时修复
// 查找权限含有 SUID 或 SGID 的文件，-o 表示 or
[root@localhost~]#find / -type f -perm -4000 -o -perm -2000
// 设置过 SUID 或者 SGID 的文件在运行过程中将会获得 root 权限，对系统安全威胁很大，经常检查含有“s”位的文件及时根据实际情况进行修复可以防止用户滥用权限或提升权限的可能性
// 检查系统中无有效属主或属组的目标
[root@localhost ~]# find / -nouser -o -nogroup
// 在找到这些文件后，要么删除，要么修改文件的属主，使其处于安全状态
```

子任务 1.3.3 /tmp、/var/tmp、/dev/shm 安全设定

Linux 为软件安装或系统运行提供了临时性文件存储目录，来保证工作顺利进行，较早的 Linux 系统一般是 /tmp 和 /var/tmp，稍微新一些的 Linux 可能还有 /dev/shm。

临时文件目录允许任何用户随时执行任何操作，为使用系统的用户提供方便，但同时也为系统留下了安全隐患。如果将病毒或者木马脚本放到临时文件目录下，那将严重影响服务器的安全。因此，临时文件目录的权限要进行安全设定。

笔记

/dev/shm 是 Linux 系统中一种基于内存的 tmpfs 文件系统，启动时自动加载。它会直接利用内存或 swap 分区来实现文件的存储，因此速度非常快。但是，直接操控系统内存这是非常危险的，所以保证 /dev/shm 安全至关重要。/tmp 在安装系统时应提前考虑到安全问题，作为一个独立磁盘分区进行安装，这样的话，/var/tmp 就可以使用软链接来链接到 /tmp 下保证安全访问了。

说明

nosuid、noexec、nodev 选项表示在这个分区不允许执行任何脚本、不能存在设备文件等。

```
[root@localhost ~]# vi  /etc/fstab  // 打开文件系统主配置文件进行编辑
/tmp  /tmp  ext3  rw,nosuid,noexec,nodev  0 0
tmpfs  /dev/shm  tmpfs  defaults,nosuid,noexec,rw  0 0
[root@localhost ~]# mount -a  // 重新挂载 fstab 配置信息
[root@localhost ~]# mv  /var/tmp/* /tmp  // 移除 /var 下内容到 /tmp
[root@localhost ~]# ln -s  /tmp /var/tmp  // 设置软链接
///tmp 往往被安装成根目录下的一个目录，那么设置稍微复杂一些。
[root@localhost ~]# dd if=/dev/zero of=/dev/tmpfs bs=1M count=10000
[root@localhost ~]# mke2fs -j /dev/tmpfs
[root@localhost ~]# cp -av /tmp /tmp.old
[root@localhost ~]# mount -o loop,noexec,nosuid,rw /dev/tmpfs /tmp
[root@localhost ~]# chmod 1777 /tmp
[root@localhost ~]# mv -f /tmp.old/* /tmp/
[root@localhost ~]# rm -rf /tmp.old
[root@localhost ~]# vi /etc/fstab  // 编辑 fstab 文件
/dev/tmpfs  /tmp  ext3  loop,nosuid,noexec,rw  0 0
[root@localhost tmp]# vi  test.sh  // 创建 shell 脚本检验设置效果
ls  -al  // 内容自定，单击“Esc”，输入“:wq”保存退出
[root@localhost tmp]# ./test.sh  // 拒绝执行
-bash: ./test.sh: Permission denied
```

可以看出，虽然文件有可执行属性，但是在 /tmp 分区已经无法执行任何文件了。

任务 1.4 系统软件安全管理

系统软件的漏洞是导致系统不安全的主要因素，因此，软件的安全下载、安装、升级等操作是保证系统安全的首要任务。如何实现软件的安全管理，首先需要了解 Linux 下的 yum 工具。

子任务 1.4.1 认识软件自动升级工具 yum

与 apt 类似，yum 是 Yellow dog Updater, Modified 的简称，是杜克大学为了提高 RPM 软件包安装性而开发的一种软件包管理器。起初是由 yellow dog 这一发行版的开发者 Terra Soft 研发，用 python 写成，那时还叫做 yup(Yellow dog Updater)，后经杜克大学的 Linux@Duke 开发团队进行改进，遂有此名。yum 的宗旨是自动化升级、安装、移除 rpm 包，收集 rpm 包的相关信息，检查依赖关系并自动解决。yum 的关键之处是配置可靠的

repository，即软件仓库（http 或 ftp 站点，也可以是本地磁盘，但必须包含 rpm 的 header，header 包括了 rpm 包的各种信息，包括描述、功能、提供的文件、依赖性等），通过互联网去下载指定的软件实现自动安装。

笔记

yum 的使用主要通过配置文件 /etc/yum.conf 来实现自动解决增加或删除 rpm 包时遇到的依赖性问题，而 yum 的库资料配置位于 /etc/yum.repos.d 目录下的各文件中。

```
[root@localhost ~]# vi /etc/yum.conf
[main]
cachedir=/var/cache/yum
//yum 的缓存目录，存储下载的 rpm 包和数据库的默认路径
keepcache=0
// 安装完成后是否保留软件包，0 为不保留（默认为 0），1 为保留
debuglevel=2
//Debug 信息输出等级，范围为 0-10，缺省为 2
logfile=/var/log/yum.log
//yum 日志文件位置。
pkgpolicy=newest
// 设置安装 rpm 包策略。newest 和 last，主配置文件中声明了多个库地址，而同一软件在不同的库地址中同时存在时，yum 会按照这个选项进行选择安装。newest，表示安装最新版本；last，选择最后地址中软件进行安装
distroverpkg=redhat-release
// 指定一个软件包，yum 会根据这个包判断你的发行版本
tolerant=1
// 有 1 和 0 两个选项，表示 yum 是否容忍命令行发生与软件包有关的错误
exactarch=1
// 有 1 和 0 两个选项，设置为 1，yum 只会安装和系统架构匹配的软件包
retries=6
// 网络连接发生错误后的重试次数，如果设为 0，则会无限重试
obsoletes=1
// 这是一个 update 的参数，具体请参阅 yum(8)，简单地说就是相当于 upgrade，允许更新陈旧的 RPM 包
plugins=1
// 是否启用插件，默认 1 为允许，0 表示不允许
exclude=selinux*
// 排除某些软件在升级名单之外
gpgcheck=1
// 有 1 和 0 两个选择，分别代表是否进行 gpg(GNU Private Guard) 校验，以确定 rpm 包的来源是有效和安全的
```

笔记

子任务 1.4.2 yum 的安装与配置

一般情况下 yum 默认已经安装，也可以通过 wget 从网上下载相关软件包进行安装，或者通过挂载系统安装光盘进行安装。

```
[root@localhost ~]# rpm -qa|grep yum  // 检测 yum 是否安装
[root@localhost ~]# mount /dev/cdrom  /mnt/cdrom/  // 挂载系统安装光盘
[root@localhost ~]# rpm -ivh yum-*.noarch.rpm  // 根据依赖提示进行安装
[root@localhost ~]# yum -v  // 检测安装是否成功
```

yum 基础安装包含有：

yum：RPM 安装 / 升级包。

yum-fastestmirror：Yum 快速镜像安装插件。

yum-metadata-parser：Yum metadate 解析器。

提示

CentOS-Base.repo 是 yum 网络源的配置文件，CentOS-Media.repo 是 yum 本地源的配置文件。

```
[root@localhost ~]# cd /etc/yum.repos.d  // 进入 yum 库文件目录
[root@localhost ~]# vi  /etc/yum.repos.d/CentOS-Base.repo
// 直接编辑库文件实现成功下载
[base] #serverid 用于区别各个不同的 repository，必须有一个独一无二的名称；
// 软件包下载基本配置信息
name=CentOS-$releasever - Base #name 是对 repository 的描述，可以包含变量
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
```

指定 yum 在互联网中下载软件包的 URL 地址：

$releasever：代表发行版的版本，从 [main] 部分的 distroverpkg 获取，如果没有，则根据 redhat-release 包进行判断。

$arch：cpu 体系，如 i686、athlon 等

$basearch：cpu 的基本体系组，如 i686 和 athlon 同属 i386，alpha 和 alphaev6 同属 alpha。

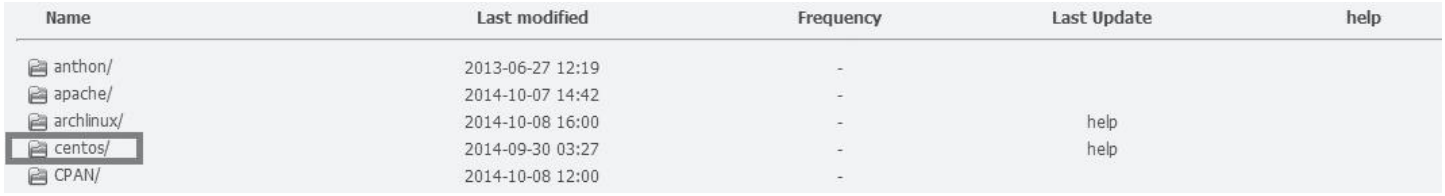
```
baseurl=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os
或 file:///media/CentOS/
//baseurl 是服务器设置中最重要的部分，只能有一个，该变量设置正确才能成功获取软件。
gpgcheck=1  // 是否启用 gpg 检查，1 表示启用，0 表示不启用校验。-RPM-KEY 的位置。
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6  // 指定 GPG 密钥的地址。
[updates]
//updates，更新模块要用到的配置
name=CentOS-$releasever - Updates
```

```
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

笔记

也可以直接下载 *.repo 文件保存到 /etc/yum.repos.d/ 目录下，无须修改直接使用，但 GPGkey 文件需要导入。

打开浏览器，输入：http://mirrors.sohu.com/，单击适合自己系统的“ Centos”进入下载页。如图 1-2 所示。



Name	Last modified	Frequency	Last Update	help
anthon/	2013-06-27 12:19	-		
apache/	2014-10-07 14:42	-		
archlinux/	2014-10-08 16:00	-	help	
centos/	2014-09-30 03:27	-	help	
CPAN/	2014-10-08 12:00	-		

图 1-2　centos 下载页

根据提示选择安装包：

```
[root@localhost ~]# rpm --import RPM-GPG-KEY-CentOS-5  // 导入 GPGkey
```

子任务 1.4.3 掌握 yum 的特点与基本用法

yum 作为前端软件包管理器，能够根据主配置文件从指定的服务器自动下载 RPM 包并执行安装，最大的好处是自行处理依赖关系。因此，熟练使用 yum 执行系统软件的查找、安装、卸载、升级等操作是保证系统安全的一个重要步骤。

语法格式：

yum [options] [command] [package ...]

部分参数：

[options]：可选，具体如下。

- -h：帮助。
- -y：当安装过程提示选择全部为“yes”。
- -q：不显示安装的过程。

[command]：进行的操作，具体如下。

- install：安装。
- update：更新。
- upgrade：升级。
- list：显示列表。
- remove：移除。
- clean：清除缓存。

[package ...]：下载的包名。

例如：

笔记

```
[root@localhost ~]# yum install dhcp  //使用 yum 实现网络安装 dhcp
[root@localhost ~]# yum update dhcp  //更新指定程序包 dhcp
[root@localhost ~]# yum check-update  //检查可更新的程序
[root@localhost ~]# yum upgrade dhcp  //升级指定程序包 package1
[root@localhost ~]# yum list  //显示所有已经安装和可以安装的程序包
[root@localhost ~]# yum list dhcp  //显示 dhcp 的安装情况
[root@localhost ~]# yum deplist dhcp  //查看程序 dhcp 依赖情况
[root@localhost ~]# yum remove dhcp  //删除程序包 dhcp
[root@localhost ~]# yum clean packages  //清除缓存目录下的软件包
[root@localhost ~]# man  yum  //查看 yum 帮助手册
```

知识拓展

yum 源介绍

Linux 系统受到越来越广泛的关注，使用的人数在快速增加，对于在 Linux 上运行的软件需求也逐步提高，包括软件的种类、软件的更新速度等。因此，yum 已经成为 CentOS 系统进行软件管理必不可少的一款工具，但是官方提供的 yum 源中软件种类并不丰富，因此针对 Linux 的运维人员来说了解到更多好用的 yum 源是方便使用 yum 的前提。

1. EPEL

EPEL 全称是企业版 Linux 附加软件包，是由特别兴趣小组创建、维护并管理的，针对红帽企业版 Linux(RHEL) 及其衍生发行版（例如 CentOS、Scientific Linux）的一个高质量附加软件包项目。其官方网址为 http://fedoraproject.org/wiki/EPEL/zh-cn。EPEL 的软件包不会与企业版 Linux 官方源中的软件包发生冲突，或者互相替换文件，因此可以放心使用。

EPEL 包含一个名为“epel-release”的包，这个包含了有 EPEL 源的 GPGKey 和软件源信息。可以通过 yum 命令将这个软件包安装到企业级 Linux 发行版上，这样就可以使用最全面、最稳定的 Linux 软件包了。

2. RPMForge

RPMForge 是一个第三方的软件源仓库，也是 CentOS 官方社区推荐的第三方 yum 源，它为 CentOS 系统提供了超过 10 000 个软件包，被 CentOS 社区认定为最安全也最稳定的一个软件仓库。但是由于这个安装源不是 CentOS 本身的组成部分，因此要使用 RPMForge，需要手动下载并安装。RPMForge 的官方网站是 http://repoforge.org/ ，可以在 http://pkgs.repoforge.org/rpmforge-release/ 下载 RHEL/Centos 各个版本的“rpmforge-release”包，这样就可以使用 RPMForge 中的丰富软件了。

任务 1.5 Linux 后门入侵检测工具

子任务 1.5.1 认识 Rootkit

获取他人计算机控制权有很多种方法，比如各种登录 login、SSH 等，但是因为有输入口令的次数限制，因此不大流行。另外还可以通过编写程序进行漏洞测试，从而利用漏洞进行攻击。其中，Rootkit 是 Linux 平台下最常见的一种危害性极大的木马后门工具，它入侵后可以获得管理员 root 权限，通过替换系统文件来达到入侵和隐蔽的目的。这种木马

比普通木马后门更加危险和隐蔽，普通的检测工具和检查手段很难发现这种木马。Rootkit 攻击能力极强，对系统的危害很大，它通过一套工具来建立后门和隐藏行迹，从而让攻击者保住权限，以使它在任何时候都可以使用 root 权限登录系统。

笔记

Rootkit 主要有两种类型：文件级别和内核级别。

1. 文件级别 Rootkit

文件级别的 Rootkit 一般通过程序漏洞或者系统漏洞进入系统后，通过修改系统的重要文件来达到隐藏自己的目的。在系统遭受 Rootkit 攻击后，合法的文件被木马程序替代，变成了外壳程序，而其内部是隐藏着的后门程序。通常容易被 Rootkit 替换的系统程序有 login、ls、ps、ifconfig、du、find、netstat 等，其中 login 程序是最经常会被替换的，因为当访问 Linux 时，无论是通过本地登录还是远程登录，/bin/login 程序都会运行，系统将通过 /bin/login 来收集并核对用户的账号和密码，而 Rootkit 就是利用这个程序的特点，使用一个带有根权限后门密码的 /bin/login 来替换系统的 /bin/login，这样攻击者通过输入设定好的密码就能轻松进入系统。此时，即使系统管理员修改 root 密码或者清除 root 密码，攻击者还是一样能通过 root 用户登录系统。通常攻击者在进入 Linux 系统后，会进行一系列的攻击动作，最常见的是安装嗅探器收集本机或者网络中其他服务器的重要数据。在默认情况下，Linux 中也有一些系统文件会监控这些工具动作，例如 ifconfig 命令，所以，攻击者为了避免被发现，会想方设法替换其他系统文件，常见的就是 ls、ps、ifconfig、du、find、netstat 等。如果这些文件都被替换，那么在系统层面就很难发现 Rootkit 已经在系统中运行了。

这就是文件级别的 Rootkit，对系统维护威胁很大，目前最有效的防御方法是定期对系统重要文件的完整性进行检查，如果发现文件被修改或者替换，那么很可能系统已经遭受了 Rootkit 入侵。检查文件完整性的工具很多，常见的有 Tripwire、aide 等，可以通过这些工具定期检查文件系统的完整性，以检测系统是否被 Rootkit 入侵。

2. 内核级别的 Rootkit

内核级别 rootkit 是比文件级别 rootkit 更高级的一种入侵方式，它可以使攻击者获得对系统底层的完全控制权，此时攻击者可以修改系统内核，进而截获运行程序向内核提交的命令，并将其重新定向到入侵者所选择的程序并运行此程序，也就是说，当用户要运行程序 A 时，被入侵者修改过的内核会假装执行 A 程序，而实际上却执行了 B 程序。

内核级别 Rootkit 主要依附在内核上，它并不对系统文件做任何修改，因此一般的检测工具很难检测到它的存在，这样一旦系统内核被植入 Rootkit，攻击者就可以对系统为所欲为而不被发现。目前针对内核级的 Rootkit 还没有很好的防御工具，因此，做好系统安全防范就非常重要，将系统维持在最小权限内工作，只要攻击者不能获取 root 权限，就无法在内核中植入 Rootkit。

子任务 1.5.2　Rootkit 后门检测工具 chkrootkit

Rootkit 是利用漏洞进入系统的，尽量做到删除没必要的用户（子任务 1.1.1）、关闭不必要的服务（子任务 1.1.2）、不要在网络上使用明文传输密码如 telnet，从而避免入侵以及入侵的蔓延。只有网络非常安全，让攻击者无隙可乘，才能使自己免受 Rootkit 的影响。因此，运维人员在日常的网络管理维护中要保持一些良好的习惯，做好对后门入侵行为的

笔记

检测。

登录 http://www.chkrootkit.org/download/，选择 tar 包进行下载，如图 1-3 所示。

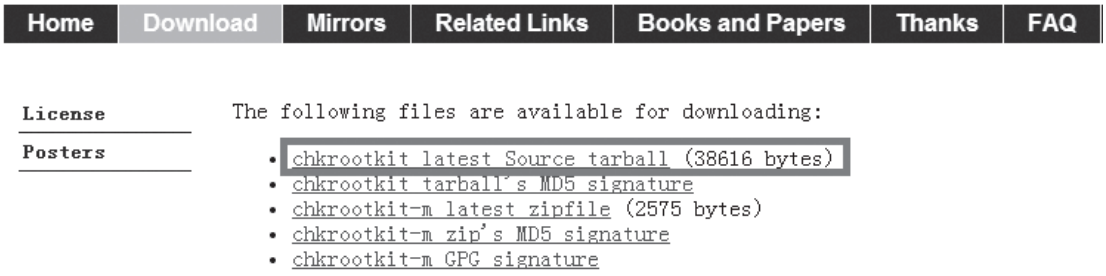


图 1-3　tar 包下载页

```
[root@localhost~]#
wegt ftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz
[root@localhost ~]# tar -zxvf chkrootkit.tar.gz
[root@localhost ~]# ls
[root@localhost ~]# cd chkrootkit-0.52
[root@localhost chkrootkit-0.52]# make sense  // 编译
[root@localhost chkrootkit-0.52]# cd ..
[root@localhost ~]# cp -r chkrootkit-0.52 /usr/local/chkrootkit
[root@localhost ~]#rm -rf chkrootkit-0.52
[root@localhost ~]# cd /usr/local/chkrootkit
[root@localhost chkrootkit]# ./chkrootkit
Checking `ifconfig'... INFECTED
Checking `ls'... INFECTED
Checking `login'... INFECTED
Checking `sshd'... not infected
```

从输出结果可以看出，带有“ INFECTED”标志的即为被感染命令，如 ifconfig、ls、login 等。针对被感染的 Rootkit 的系统，最安全而有效的方法就是备份数据并重新安装系统。Chrootkit 在检测过程中使用了部分系统命令，如果服务器被黑客入侵，那么这些系统命令也可能被入侵者替换，此时 chkrootkit 的检测结果将变得完全不可信。为了避免这个问题，先将 chkrootkit 使用的系统命令进行备份，在需要的时候使用备份的原始系统命令让 chkrootkit 对 Rootkit 进行检测。

```
[root@localhost ~]# mkdir /usr/share/.commands  // 创建隐含目录
[root@localhost ~]# cp `which --skip-alias awk cut echo find egrep id head ls
netstat ps strings sed uname` /usr/share/.commands
// 将命令转存到隐含目录
[root@localhost ~]# /usr/local/chkrootkit/chkrootkit -p /usr/share/.commands/
//-p 指定路径使用系统命令
[root@localhost ~]# cd /usr/share/  // 进入目录
```

```
[root@localhost share]# tar -zcvf commands.tar.gz  .commands
// 将隐含目录归档压缩
[root@localhost share]# cp  commands.tar.gz  /mnt/command
// 将文件备份到其他目录
```

笔记

子任务 1.5.3 Rootkit 后门检测工具 RKHunter

RKHunter 是 Linux 下的一款开源入侵检测工具，是专业检测系统是否感染 Rootkit 的一个工具，它比 chkrootkit 更为全面，除了 Rootkit 特征码扫描外，还支持端口扫描，它通过执行一系列的脚本来确认服务器是否已经感染 Rootkit。

登录 https://sourceforge.net/projects/rkhunter/files/ 下载安装包 rkhunter-1.4.2.tar.gz 。

```
[root@localhost ~]# tar -zxvf rkhunter-1.4.2.tar.gz
[root@localhost ~]# ls
[root@localhost ~]# cd rkhunter-1.4.2
[root@localhost rkhunter-1.4.2]# ./installer.sh  // 查看帮助
[root@localhost rkhunter-1.4.2]# mkdir /usr/local/rkhunter  // 创建安装目录
[root@localhost rkhunter-1.4.2]# ./installer.sh  --layout custom  \
  /usr/local/rkhunter --install
[root@localhost rkhunter-1.4.2]# cd  /usr/local/rkhunter
[root@localhost rkhunter]# ls
[root@localhost rkhunter]# cd bin
[root@localhost rkhunter]# ls
[root@localhost bin]#./rkhunter  --help  // 查看帮助信息
[root@localhost rkhunter]# ./rkhunter  -c  // 分成几个部分显示检测信息
....
Checking system commands...
.....
[Press <ENTER> to continue]  // 单击“回车”进入下一部分检测信息
// 检测常见的 rootkit 程序，显示“Not found”表示系统未感染此 rootkit
Checking for rootkits...
Performing check of known rootkit files and directories
.......
[Press <ENTER> to continue]
// 特殊或附加的检测，例如对 rootkit 文件或目录检测、对恶意软件检测以及对指定的内核模块检测
Performing additional rootkit checks
.....
[Press <ENTER> to continue]
```

笔记

```
    // 对网络、系统端口、系统启动文件、系统用户和组配置、SSH 配置、文件系统等进行检测
    Checking the network...
    Performing checks on the network ports
    ......
    [Press <ENTER> to continue]
    // 对应用程序版本进行检测
    Checking application versions...
    ......
    // 输出总结信息
    System checks summary
    =====================
    File properties checks...
    ......
    The system checks took: 8 minutes and 21 seconds
    // 每项检测结果都使用不同的颜色显示，绿色表示没有问题；红色表示严重
    [root@localhost ~]# crontab  -e  // 为当前用户编辑 crontab 任务计划配置文件
    // 分（0-59） 小时（0-23） 日（1-31） 月（1-12） 星期（0-6）命令
    30  8  *  *  *  /usr/local/rkhunter/rkhunter --check --cronjob
    // 每天的 8:30 作为一个 cron 运行一次。
    [root@localhost ~]# /sbin/service crond start  // 开启 cron 服务使配置生效
```

任务 1.6 服务器遭受攻击后的处理过程

再安全的服务器也有可能遭受攻击，运维人员要把握的原则是：尽量做好系统安全防护，修复所有已知的危险行为，同时，在系统遭受攻击后能够迅速有效地处理攻击行为，最大限度地降低攻击对系统产生的影响。常见处理方法有以下几种。

（1）断开网络。网络是所有攻击的源头。遭受攻击后，首先断开服务器的网络连接，将服务暂停至少 3 小时，进行服务器维护。既保证了及时切断攻击源，又能保护服务器所在网络的其他主机。

（2）分析日志文件。对服务器进行全盘杀毒，检测后门入侵程序是否存在（如 chrootkit）；分析服务器日志文件记录，查看可疑信息（如可疑用户遗留下的 IP 地址等）；查看系统开启的端口、运行的进程等，分析可疑程序。

（3）设置更严格的权限。为服务器设置更严格的访问权限控制，关闭不必要的服务，删除不必要的账户，重新设置防火墙规则，更改密码，检测系统漏洞、程序漏洞，确定入侵原因；查找攻击途径，确定攻击源。

（4）备份用户数据。检查用户数据是否已被入侵，经过查杀检测后立刻备份服务器上的用户数据。

（5）重新安装系统。无论经过怎样的安全检测步骤，都无法保障系统彻底干净，因此，最安全也最简单的方法就是重装系统。

（6）修复漏洞恢复服务。修复漏洞程序，将备份的数据重新复制到新安装系统上，开启服务，开启网络连接，提供服务。

笔记

子任务 1.6.1 检查并锁定可疑用户

服务器遭受攻击总是来自于可疑用户登录后的非法操作，因此，首先查看系统中有哪些可疑用户，一旦发现立刻将该用户锁定，然后中断此用户的远程连接。

1. 检测可疑账号并锁定剔除

```
[root@localhost ~]# w  // 显示系统登录账号信息查找可疑用户
[root@localhost ~]# passwd -l nobody
// 锁定可疑用户（/etc/passwd 默认不允许 nobody 登录）
[root@localhost ~]# ps -ef  // 查看进程执行情况
UID   PID   PPID   C   STIME   TTY   TIME   CMD
531   6051   6049   0   19:23   ?   00:00:00 sshd:nobody@pts/3
[root@localhost ~]# kill -9 6051  // 将用户执行的 PID 强制杀死。
```

2. 使用 last 进行非法账号检测

语法格式：last [-R][-n][-f file][-t tty][-h 节点][-i IP][-1][-y][ID]

参数及其含义：

-R：省略 hostname。

-n：指定输出记录的条数。

-f file：指定查询 log 文件。

-t tty：指定查找终端。

-h 节点：指定节点显示登录信息。

-i IP：指定 IP 显示信息。

-l：显示远端地址。

-y：显示记录的年月日。

ID：指定查询的用户名。

Last 命令是显示近期用户或终端的登录情况，它主要是通过访问 /var/log/wtmp 文件来获得信息的。Wtmp 文件是一个日志文件，该文件记录了每个用户的登录、注销以及系统的启动 / 关闭等。

```
[root@localhost ~]# last -R  -5
```

子任务 1.6.2 查看系统日志

系统日志能记录下很多重要信息，如服务启动、运行过程中的错误信息，因此，当系统受到意外攻击后，查看日志是定位攻击源最好的方法。运维人员必须了解系统中常见日志名称及其作用。

Linux 系统通过不同的守护进程，为管理员自动记录了很多种日志。

说明

UID：用户 ID，输出执行该进程的用户名。

PID：进程 ID。

PPID：父进程 ID。

C：进程占用 CPU 的百分比。

STIME：进程启动到现在的时间。

TTY：该进程在哪个终端上运行，若与终端无关，显示“?”；若为 pts/0 等，则表示是网络连接执行的进程。

TIME：该进程使用 CPU 运行时间。

CMD：进程名。

笔记

```
[root@localhost ~]# cat  /var/log/dmesg  // 核心启动日志
[root@localhost ~]# cat  /var/log/boot.log  // 系统的引导日志
[root@localhost ~]# cat  /var/log/messages  // 系统报错日志
[root@localhost ~]# cat  /var/log/maillog  // 邮件系统日志
[root@localhost ~]# cat  /var/log/xferlog  // FTP 系统日志
[root@localhost ~]# cat  /var/log/secure  // 存储来自于认证模块 PAM 的日志，如：安全信息、系统登录、网络连接认证方式等，常见于 Centos
[root@localhost ~]# cat  /var/log/wtmp  // 登录信息日志
[root@localhost ~]# cat  /var/log/rpmpkgs  // RPM 软件包管理日志
[root@localhost ~]# cat  /var/log/cron  // 任务计划日志
[root@localhost ~]# cat  ~/.bash_history  // 记录用户执行过的所有历史命令
```

子任务 1.6.3 检查并关闭系统可疑进程

系统遭到入侵后，及时检查可疑进程并确定攻击源，是一个运维人员必备的技能。Linux 系统提供了很多用于检测进程的相关命令，如 ps、top 等。

1. 查找指定名称的进程 id 号

语法格式：pidof [选项] 进程名称

选项：

-s：仅返回一个进程号。

-x：显示由脚本开启的进程。

-o：指定不显示的进程 id。

```
[root@localhost ~]# pidof sshd
13276 12942 4284
[root@localhost ~]# ls -al /proc/13276/exe  // 查看对应 PID 目录下的 exe 文件信息
lrwxrwxrwx 1 root root 0 Oct  4 22:09 /proc/13276/exe -> /usr/sbin/sshd
```

确定进程的完整执行信息，帮助系统运维人员准确关闭可疑进程。

2. 显示所有正在使用指定文件、文件系统或套接字的进程信息

语法格式：fuser[-option] 文件

```
[root@localhost ~]# fuser /etc/passwd  // 列出正在使用 passwd 文件的进程 id 号
[root@localhost ~]# fuser -u /etc/passwd  // 列出正在使用 passwd 文件的进程号和用户名
```

3. 通过发送指定信息给进程来实现终止进程

语法格式：kill [-s 信息名称或编号] PID

```
[root@localhost ~]# kill  PID  // 将上面命令显示出来的 PID 写入命令
[root@localhost ~]# kill -9  PID  //-9 表示强制
```

笔记

子任务 1.6.4 检查文件系统的完好性

被攻击过的系统中系统命令也常常被恶意替换，替换过的文件在其属性上会有所变化，所以通过检查文件属性是否发生变化也可以验证文件系统是否被入侵。Linux 系统中 rpm 命令提供了包校验功能，该功能可以通过比较软件包中安装的文件信息和软件包中原始文件的信息是否相同，来确定文件属性是否发生改变。包校验主要包括：比较文件的长度、MD5 校验、许可、类型、文件属主和群组等属性。

```
[root@localhost ~]# rpm -Va
```

-V 表示进行包校验，-a 表示所有安装的 rpm 包输出结果，如图 1-4 所示。

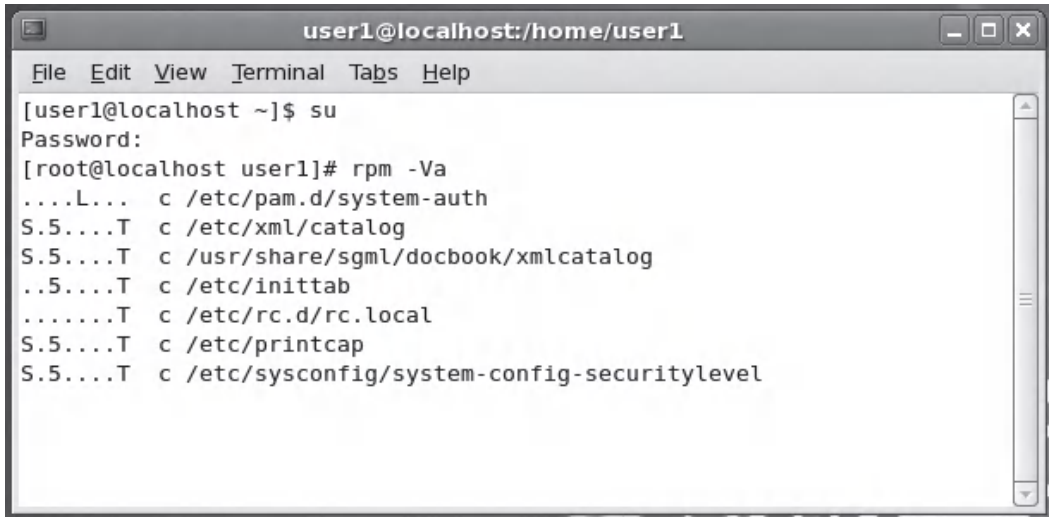


图 1-4　rpm 检测结果

输出中显示各标记及其含义：

S：表示文件长度发生了变化。

M：表示文件的访问权限或文件类型发生了变化。

5：表示 MD5 校验发生了变化。

D：表示设备节点的属性发生了变化。

L：表示文件的符号链接发生了变化。

U：表示文件 / 子目录 / 设备节点的 owner 发生了变化。

G：表示文件 / 子目录 / 设备节点的 group 发生了变化。

T：表示文件最后一次的修改时间发生了变化。

如果在输出结果中有“M”标记出现，那么对应的文件可能已经遭到篡改或替换，此时可以通过卸载这个 RPM 包重新安装来清除受攻击的文件。这种方法仅限于检查 rpm 包的安装软件，其他方法安装的软件不包括在内。

```
[root@localhost ~]# rpm  -e  包名
```

笔记

任务 1.7 Linux 入侵分析

子任务 1.7.1 了解受攻击现象

一台门户网站服务器，持续对外发送数据包，导致 100 MB 带宽耗尽，服务器安装 CentOS 5.5，对外开放 80、22 端口。

初步了解，网站的访问量不大，带宽占用也不太高，而耗尽 100 MB 的带宽是绝对不可能的，那么极有可能是服务器遭受了流量攻击，因此需要登录服务器做详细的检测。

子任务 1.7.2 初步分析

通过交换机对该服务器的网络流量进行检测，发现该主机确实存在对外 80 端口的扫描流量。

```
[root@localhost ~]# netstat -an  // 检查系统端口的占用情况
[root@localhost ~]# ps -ef  // 检测进程
[root@localhost ~]# top  // 查看实时信息
```

未发现明显问题，初步怀疑系统被植入了 Rootkit，找到系统健康时备份的命令，使用 md5sum 命令检测文件的完整性。

```
[root@localhost ~]# md5sum  ps -c  ps.md5  #md5sum 检测
```

通过输出比较结果断定文件变化，确认服务器已经被入侵、且安装了 Rootkit 级别的后门程序。

子任务 1.7.3 断网分析系统

将服务器断开网络，分析系统日志，寻找攻击源。为了防止系统中的常用命令被入侵，检测结果不可信，因此，将与服务器配置相同的操作系统下的所有命令复制一份覆盖服务器命令后执行相关操作。

```
[root@localhost ~]# more /var/log/secure |grep Accepted
// 分析显示结果中可疑记录
Oct 3 03:10:25 webserver sshd[20701]: Accepted password for mail from 62.17.163.186  port 53349 ssh2
// 日志显示在 10 月 3 日凌晨 3 点 10 分，有个 mail 账号从 62.17.163.186 这个 IP 成功登录了系统，mail 是系统的内置账号，默认无法执行登录操作的。因此检验 62.17.163.186 这个 IP 地址的位置
[root@localhost ~]# vi  /etc/shadow
// 根据显示结果检测和 mail 相关的可疑信息
mail:$1$kCEd3yD6$W1evaY5BMPQIqfTwTVJiX1:15400:0:99999:7:::
```

```
// 在主配置中，mail 账号已经被设置了密码，并且被修改为可远程登录
[root@localhost ~]# cat /var/log/messages
[root@localhost ~]# cat /var/log/wtmp
[root@localhost ~]# cat /var/log/secure
[root@localhost ~]# cat ~/.bash_history
// 继续查看其他系统日志分析可疑信息
```

笔记

子任务 1.7.4　寻找攻击源

针对本次问题发现的线索：① 62.17.163.186 地址；② mail 账号曾经登录过系统，执行如下操作。

```
[root@localhost ~]# ps -aux  // 查看详细进程信息分析可疑程序
nobody  22765  ……  1  6 Sep29 ?  4-00:11:58 .t
[root@localhost ~]# top  // 针对可疑程序作进一步分析
PID USER PR NI VIRT RES SHR S %CPU %MEM  TIME+  COMMAND
22765 nobody  150 1740m 1362m 1228 S  98.3  91.5 2892:19 .t
```

该进程 PID 为 22765，已经运行了 4 天左右，运行这个程序的是 nobody 用户，并且这个 t 程序消耗了大量的内存和 CPU，这也是之前客户反映的网站服务器异常缓慢的原因。

```
[root@localhost ~]# ls -al /proc/22765/exe
lrwxrwxrwx 1 root root 0 Sep2922:09 /proc/22765/exe -> /var/tmp/…/apa/t
```

准确定位到了进程的完整程序执行路径。入侵者在 /var/tmp 目录下创建了一个“…”的目录，在这个目录下隐藏着攻击的程序源。

```
[root@localhost ~]# cd  /var/tmp/…/
[root@localhost ...]# ls -al
…
drwxr-xr-x 2  nobody nobody  4096  Sep 29 22:09 haha
-rwxr-xr-x 1  nobody nobody  0  Sep 29 22:10 login
-rw-r--r-- 1  nobody nobody  0  Sep 29 22:10 login.tgz
-rwxr-xr-x 1  nobody nobody  0  Sep 29 22:10 z
…
```

通过分析这些文件，判断攻击源程序的目的。

子任务 1.7.5　查找攻击原因

服务器安装的软件有 Apache2.0.63、Tomcat5.5，Apache 和 Tomcat 之间通过 mod_jk 模块进行集成，Aapache 对外开放 80 端口。由于 Tomcat 没有对外开放端口，因此问题在 Apache 上。

笔记

通过查看 Apache 的配置发现，Apache 仅仅处理些静态资源请求，而网页也以静态页面居多，所以通过网页方式入侵系统可能性不大。既然漏洞可能来自 Apache，那么尝试查看 Apache 日志，也许能发现一些可疑的访问痕迹。通过查看 access.log 文件，发现了如下信息：

```
62.17.163.186 - - [29/Sep/2013:22:17:06 +0800] "GET http://www.xxx.com/cgi-bin/awstats.pl?configdir=|echo;echo;ps+-aux%00 HTTP/1.0" 200 12333 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1) Gecko/20121010 Firefox/2.0"
62.17.163.186 - - [29/Sep/213:22:17:35 +0800] "GET http://www.xxx.com/cgi-bin/awstats.pl?configdir=|echo;echo;cd+/var/tmp/.../haha;ls+-a%00 HTTP/1.0" 200 1626 "-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.8.1) Gecko/20121010 Firefox/2.0"
```

至此，发现了漏洞的根源，原来是 awstats.pl 脚本中 configdir 的一个漏洞，通过了解此服务器的应用，客户确实是通过一个 Awstats 的开源插件来做网页访问统计，通过这个漏洞，攻击者可以直接在浏览器上操作服务器，例如查看进程、创建目录等。通过以上第二条日志可以看出，攻击者将正常浏览器执行切换到 /var/tmp/.../haha 目录的操作。

子任务 1.7.6 揭开谜团

此服务器遭受入侵的原因和过程已经非常清楚了，大致过程如下：

（1）攻击者通过 Awstats 脚本 awstats.pl 文件的漏洞进入系统，在 /var/tmp 目录下创建了隐藏目录，然后将 Rootkit 后门文件传到这个路径下。

（2）攻击者通过植入后门程序，获取了系统超级用户权限，进而控制了这台服务器，通过这台服务器向外发包。

（3）攻击者的 IP 地址 62.17.163.186 可能是通过代理过来的，也可能是攻击者控制的其他肉机服务器。

（4）攻击者为了永久控制这台机器，修改了系统默认账号 mail 的信息，将 mail 账号变为可登录，并且设置了 mail 账号的密码。

（5）攻击者在完成攻击后，通过后门程序自动清理了系统访问日志，销毁了证据。

通过对这个入侵过程的分析，发现入侵者的手段还是非常简单和普遍的，虽然入侵者删除了系统的一些日志，但还是留下了很多可查的踪迹。

子任务 1.7.7 恢复网站

备份系统数据文件，重新安装系统，基本步骤如下：

（1）安装稳定版本的操作系统，删除系统默认且不需要的用户。

（2）系统登录方式改为公钥认证方式，避开密码认证的缺陷。

（3）安装更高版本的 Apache 和最新稳定版本的 Awstats 程序。

（4）使用 Linux 下的 tcp_wrappers 防火墙，限制 SSH 登录的源地址。

Rootkit 后门入侵攻击是 Linux 系统下比较常见的一种攻击手段，虽然此类攻击比较难以防范，但是目前已经有很多可以检测 Rootkit 的工具。

项目 2 Linux 网络安全运维

项目目标

1. 了解网络安全知识。
2. 了解常见 Internet 安全隐患。
3. 熟悉网络攻击手段和策略。
4. 掌握安全配置工具的使用。

知识导图

笔记

项目导入

随着现代通信技术及计算机网络的迅速发展，Internet 全球信息网在世界各个领域中发挥着日益重要的作用，人们也越来越多地依赖 Internet 进行信息管理、信息共享及经济贸易活动。但随着 Internet 使用范围的不断扩大与用户的不断增加，网络的安全性已越来越严重，安全机制也受到越来越多的关注，同时也面临着各种各样的威胁和攻击。近年来侵袭计算机专用网络的事件迅速增多，几乎与 Internet 用户的迅速增长同步，造成了大量的信息丢失和严重的经济损失。网络安全问题已经引起了人们的极大关注，因此必须了解 Internet 上存在的安全隐患与风险，并能及时针对问题进行合理的修复。

网络信息既有存储在网络节点上的信息资源，即静态信息。又有传播于网络节点间的信息资源，即动态信息。而在这些信息中有些是开放的，有些是私密的。从技术角度分析，Internet 上存在以下安全隐患：

（1）缺乏安全防范措施。

（2）系统不完备的安全配置。

（3）不完善的系统管理。

（4）协议存在的安全问题。

（5）网络提供不完备的服务程序。

当系统接入 Internet 后，就面临着巨大的安全威胁，作为服务系统的运维人员，熟悉一些黑客常用的攻击手段和解决策略，对于维护自身网络安全和系统安全是大有裨益的。

任务 2.1 网络实时流量监测工具 iftop

子任务 2.1.1 初识 iftop

随着信息技术的广泛使用，人们对于网络的依赖程度越来越深，生活中几乎处处都在使用网络，网速成了人们最关心的一件事。快速的网络是服务器提供服务的有力保障。作为运维人员要实时监控系统的流量使用情况，来及时进行系统流量的最优化调整。

iftop 是 Linux 系统提供的一个流量监测命令，提供图形化界面来实时显示带宽的使用情况。它可以监控指定网卡的实时流量、端口连接信息、反向解析 IP 等，还可以精确显示本机网络流量情况及网络内各主机与本机相互通信的流量集合，非常适合于监控代理服务器或路由器的网络流量。同时，iftop 对检测流量异常的主机非常有效，通过 iftop 的输出可以迅速定位主机流量异常的根源，这对于网络故障排查、网络安全检测是十分有用的。

子任务 2.1.2 安装 iftop

系统默认没有提供 iftop 的安装，不过需要使用 iftop 是可以自行安装的。http://www.ex-parrot.com/pdw/iftop/download/ 上提供了各种不同版本的源码安装包，推荐下载最新版本。Iftop 的安装依赖很多其他软件，需要提前准备，并且一定要配置好 yum 源的支持。

```
[root@localhost ~]# cd /etc/yum.repos.d    // 进入 yum 源目录备份旧源
[root@localhost yum.repos.d]# mv CentOS-Base.repo CentOS-Base.repo.backup
```

```
    // 下载 163 的 yum 源配置文件，放入 /etc/yum.repos.d/
    [root@localhost yum.repos.d]# wget http://mirrors.163.com/.help/CentOS6-Base-163.repo
    [root@localhost yum.repos.d]# yum makecache  // 生成缓存
    [root@localhost yum.repos.d]# yum -y update  // 更新系统
    [root@localhost yum.repos.d]# cd  // 退出当前目录返回宿主目录，安装依赖软件
    [root@localhost ~]# yum install  libpcap libpcap-devel ncurses ncurses-devel
    // 网络下载 iftop 安装包
    [root@localhost ~]#wget http://www.ex-parrot.com/pdw/iftop/download/iftop-0.17.tar.gz
    [root@localhost ~]# ls
    [root@localhost ~]# tar -zxvf iftop-0.17.tar.gz  // 解压文件
    [root@localhost ~]# cd iftop-0.17  // 进入安装包目录
    [root@localhost iftop-0.17]# ./configure  --prefix=/usr/local/iftop
    // 配置安装环境
    [root@localhost iftop-0.17]]# make  & make install  // 编译并安装
```

iftop 工具安装结束。

子任务 2.1.3 使用 iftop 监控网卡实时流量

语法格式：iftop [-option]

参数及其含义：

-i：设定监测的网卡。

-B：以 bytes 为单位显示流量（默认是 bits）。

-n：使 host 信息默认直接都显示 IP。

-N：使端口信息默认直接都显示端口号。

-F：显示特定网段的进出流量。

-h：显示参数信息。

-p：使用这个参数后，中间的列表显示的本地主机信息，出现了本机以外的 IP 信息。

-b：使流量图形条默认显示。

-f：过滤计算包时使用。

-P：使 host 信息及端口信息默认显示。

-m：设置界面最上边的刻度的最大值，刻度分五个大段显示。

```
    [root@localhost ~]# cd /usr/local/iftop  // 进入 iftop 的安装目录下
    [root@localhost iftop]# ls  // 查看安装记录
    [root@localhost iftop]# cd sbin  // 进入命令所在目录
    [root@localhost sbin]# ./iftop -P -i eth1  // 显示 eth1 网卡的流量使用情况
```

笔记

笔记

iftop 监控结果如图 2-1 所示，分为三部分。

第一部分显示流量刻度尺。

第二部分左侧第一列和第二列显示哪些主机正在和本机进行网络连接。其中，“ =>”代表发送数据，“ <=”代表接收数据，通过这些指示箭头可以了解两个 IP 之间的通信情况。最右侧三列分别表示外部 IP 连接到本机 2 s、10 s 和 40 s 内的平均流量值。

第三部分分为三行，其中，“ TX”表示发送数据，“ RX”表示接收数据，“ TOTAL”表示发送和接收的总流量。与这三行对应的有三列，其中，“ cum”列表示从运行 iftop 到目前的发送、接收和总数据流量；“ peak”列表示发送、接收以及总的流量峰值；“ rates”列表示过去 2 s、10 s、40 s 内的平均流量值。

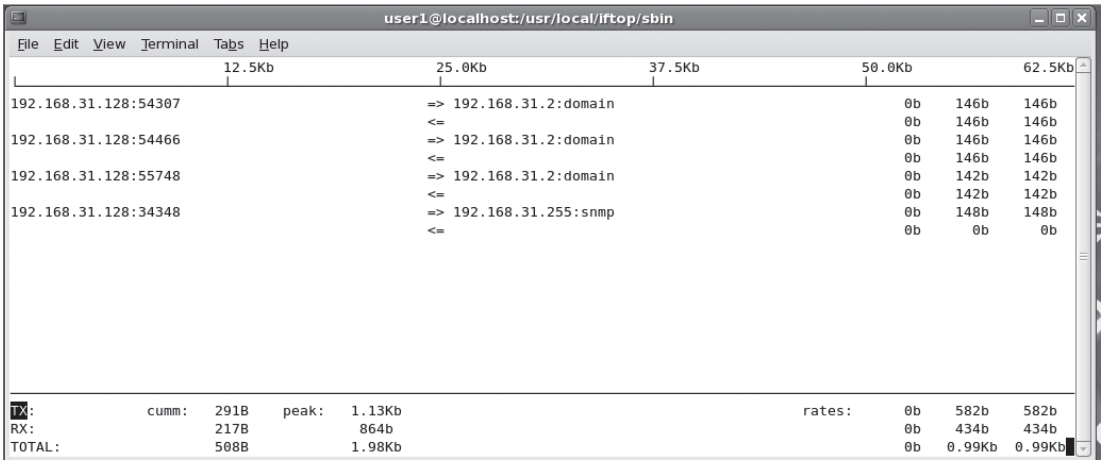


图 2-1　iftop 监控结果

知识拓展

交互模式下常用的快捷键

执行 iftop 命令后进入交互模式，交互模式下常用快捷键及其用法如下：

按 h 切换是否显示帮助；

按 n 切换显示本机的 IP 或主机名；

按 s 切换是否显示本机的 host 信息；

按 d 切换是否显示远端目标主机的 host 信息；

按 t 切换显示格式为 2 行 /1 行 / 只显示发送流量 / 只显示接收流量；

按 N 切换显示端口号或端口服务名称；

按 S 切换是否显示本机的端口信息；

按 D 切换是否显示远端目标主机的端口信息；

按 p 切换是否显示端口信息；

按 P 切换暂停 / 继续显示；

按 b 切换是否显示平均流量图形条；

按 B 切换计算 2 秒或 10 秒或 40 秒内的平均流量；

按 T 切换是否显示每个连接的总流量；

按 l 打开屏幕过滤功能，输入要过滤的字符；

按 L 切换显示画面上边的刻度；刻度不同，流量图形条会有变化；

按 j 或按 k 可以向上或向下滚动屏幕显示的连接记录；
按 1 或 2 或 3 可以根据右侧显示的三列流量数据进行排序；
按 < 根据左边的本机名或 IP 排序；
按 > 根据远端目标主机的主机名或 IP 排序；
按 o 切换是否固定只显示当前的连接；
按 f 可以编辑过滤代码；
按 ! 可以使用 shell 命令；
按 q 退出监控。

任务 2.2　网络流量监控与分析工具 Ntop 和 Ntopng

iftop 工具可以轻松实现单台服务器的流量监控，如果是针对一个庞大的服务网络，需要分析网络中每台主机以及端口的网络状态时，iftop 显然就做不到了。因此，配置一个高效的网络管理系统是必不可少的。

子任务 2.2.1 了解 Ntop 与 MRTG 的异同

MRTG 是一个监控网络链路流量负载的工具软件，通过 SNMP 协议得到设备的流量信息，并将流量负载以包含 PNG 格式图片的 html 网页形式展示给用户。MRTG 是使用 perl 语言编写的，源代码完全开放，具有高可移植性。自身有配置工具套件，使配置过程非常简单。它占用的系统资源小，可以非常直观地显示流量负载，但是只能用于 TCP/IP 网络，数据不能重复使用，无法记录更详细的流量状态，也没有管理功能。

Ntop 也是一种监控网络流量的工具，它是一种网络嗅探器，在监测网络数据传输、排除网络故障方面功能十分强大。它能自动从网络中识别有用的信息，将截获的数据包转换成易于识别的格式，通过分析网络流量来判断网络上存在的各种问题。还可以监控是否有黑客正在攻击网络，如果网络突然变缓慢，通过 Ntop 截获的数据包，可以确定是什么类型的数据包占据了大量带宽，以及数据包的发送时间、数据包传送的延时、数据包的来源地址等。用 ntop 显示网络的使用情况比其他网络管理软件更直观、详细。通过这些信息，运维人员可以及时做出响应，或者对网络进行调整，从而保证网络正常、稳定运行。

子任务 2.2.2 Ntop 与 Ntopng 的功能介绍

Ntop 提供了命令行界面和 Web 界面两种工作方式，通过 Web 界面，可以清晰展示网络的整体使用情况、网络中各主机的流量状态与排名、各主机占用的带宽以及各时段的流量明细、局域网内各主机的路由、端口使用情况等。

它从分析网络流量角度来确定网络上存在的各种问题，在 Ntop 版本更新到 Ntop5.x 之后，官方宣布停止 Ntop 版本的更新，继而推出替代版本 Ntopng。Ntopng 使用 Redis 键值服务按时间序列存储统计信息，通过这种方式实现了流量状态实时展示。与 Ntop 类似，Ntopng 也内置 Web 服务功能，同时，也支持命令行界面和 Web 界面两种工作方式，但是 Ntopng 降低了对 CPU 和内存的使用率，资源消耗更少。Ntopng 除了可以实现 Ntop 的所

笔记

有功能外，还新增了以下功能：

（1）显示 IP 流量子网矩阵。

（2）报告使用 IP 协议，按协议类型排序。

（3）生产的 HTML5/AJAX 网络流量统计。

（4）在运行时无需重启。

（5）实时监控工具汇总数据（每 5 分钟）。

（6）基于 HTML5 动态图形用户界面，分类，DPI 等。

子任务 2.2.3 安装 Ntop 与 Ntopng

Ntop 支持 UNIX（包括 Linux、BSD 和 MacOSX）、Win32（包括最新的 Windows 版本）等平台，其官方网站 http://www.ntop.org/ 提供了各种系统以及版本的下载资源，建议使用源码包实现稳定版本的安装。该软件需要提前安装依赖环境，因此，要保证 yum 源的支持。

1. 安装 Ntop

1）安装 Ntop 必需的软件环境

```
// 关闭 SElinux 防火墙以便于软件的安装（可选）
[root@localhost ~]# setenforce 0  // 临时关闭 SElinux
[root@localhost ~]# service iptables stop  // 关闭 iptables 防火墙
// 安装 ntop 的依赖环境（开发库及文件）
[root@localhost ~]# yum -y install libpcap libpcap-devel libtool libpng gdbm gdbm-devel glib libxml2-devel pango pango-devel gd zlib zlib-devel
[root@localhost ~]# yum -y install svn rrdtool rrdtool-devel python python-devel GeoIP GeoIP-devel
// 如果提示 rrdtool（一个强大的绘图引擎）法安装，可单独下载源码包执行安装。到网站 http://download.chinaunix.net/download.php?ResourceID=9078&id=35595 去下载 rrdtool-1.4.5.tar.gz
[root@localhost ~]# tar -zxvf rrdtool-1.4.5.tar.gz
[root@localhost ~]#ls
[root@localhost ~]#cd rrdtool-1.4.5
[root@localhost rrdtool-1.4.5]#./configure --prefix=/usr/local/rrdtool
[root@localhost rrdtool-1.4.5]#make & make install
```

2）编译安装 Ntop

```
[root@localhost ~]# wget
https://sourceforge.net/projects/ntop/files/ntop/ntop-4.0.1/ntop-4.0.1.tar.gz/download
// 如果提示无法实现安全连接，可根据提示为 wget 命令增加参数 --no-check-certificate 后重新执行
```

```
[root@localhost ~]# wget --no-check-certificate
https://sourceforge.net/projects/ntop/files/ntop/ntop-4.0.1/ntop-4.0.1.tar.gz/download
[root@localhost ~]# tar -zxvf ntop-4.0.1.tar.gz // 解压安装文件
[root@localhost ~]# ls  // 查看解压后安装包目录
[root@localhost ~]# cd ntop-4.0.1  // 进入安装包目录
[root@localhost ntop-4.0.1]# ./autogen.sh  -prefix=/usr/local/ntop  --with-tcpwrappers
[root@localhost ntop-4.0.1]# make  & make install
```

笔记

其中，--with-tcpwrap 选项用于支持 tcp_wrappers 访问控制，以保证 Ntop Web 访问的安全。

3）简单配置 Ntop

为了保证安全，为 Ntop 创建用户和用户组来管理 Ntop 进程。

```
[root@localhost ~]# useradd  ntop
[root@localhost ~]# passwd ntop
[root@localhost ~]# chown -R ntop:ntop /usr/local/ntop
[root@localhost ~]# mkdir /var/log/ntop
[root@localhost ~]# chown -R ntop:ntop /var/log/ntop // 赋予 ntop 写入 log 权限
```

Ntop 提供了 Web 页面来修改相关设置，如关闭 Ntop 服务等，但必须通过管理员用户的验证。Ntop 默认管理员为 admin，密码为空。因此，考虑到安全问题需要为其设置一个密码。

语法格式：ntop -P[directory] -u[user] –A

参数及其含义：

-u[user]：指定启动 service 的 user。

-A：设定 admin 密码，ntop 会内建 admin 管理者账号于 ntop 中。

-c --sticky-hosts：保留非活动主机的记录。

-P[directory]：指定数据库存放路径，即 .db 档存放路径。

-b --disable-decoders：关闭协议解码器。

-n：使用数字形式的主机标识（不做 DNS 解析）。

-w：指定监听 HTTP 访问的端口（默认为 3000），例如：

./ntop -P /usr/local/ntop/var/ntop -u ntop -w 4000(port)。

-i：指定接口名称。

4）启动运行

```
[root@localhost ~]#  cd /usr/local/ntop/bin
[root@localhost bin]# ./ntop  -A  // 设置 admin 密码
Please enter the password for the admin user: 123456
Please enter the password again: 123456
```

笔记

```
Mon Mar 27 21:47:36 2017  Admin user password has been set
Mon Mar 27 21:47:36 2017  Admin password set...
[root@localhost bin]# ./ntop -P /var/log/ntop -u ntop // 启动 ntop
```

Ntop 的 Web 页面在默认情况下没有访问限制，有时候为了网络的安全，建议设置授权访问，只有授权的主机才能访问此 Web 页面，这可以通过 Linux 系统本身的 tcp_wrappers 功能实现，授权过程如下。

```
[root@localhost ~]# vim /etc/hosts.allow
ntop: 171.22.55.21
[root@localhost ~]# vim /etc/hosts.deny
ntop: ALL
```

这里设置只允许 IP 地址为 172.22.55.21 的主机访问 Ntop 服务，禁止其他所有 IP 访问。Ntop 的安装就结束了，可以开始使用。

在完成 Ntop 安装后，执行如下命令即可启动 Ntop 服务。

```
[root@localhost ~]# ntop -i eth0 -L -d
```

这里通过 Ntop 命令监控网卡 eth0 的流量状态，在执行此命令后，Ntop 服务的日志输出将重定向到系统的 /var/log/messages 文件中。同时将开启默认的 3000 端口作为 Web 界面服务端口，打开浏览器，地址栏输入 http://IP:3000（http://192.168.31.128:3000）即可访问 Ntop 提供的 Web 监控界面。

Ntop 的 Web 界面主要由 7 个主栏目组成，下面介绍每个栏目中需要重点关注的功能点。如图 2-2 所示。

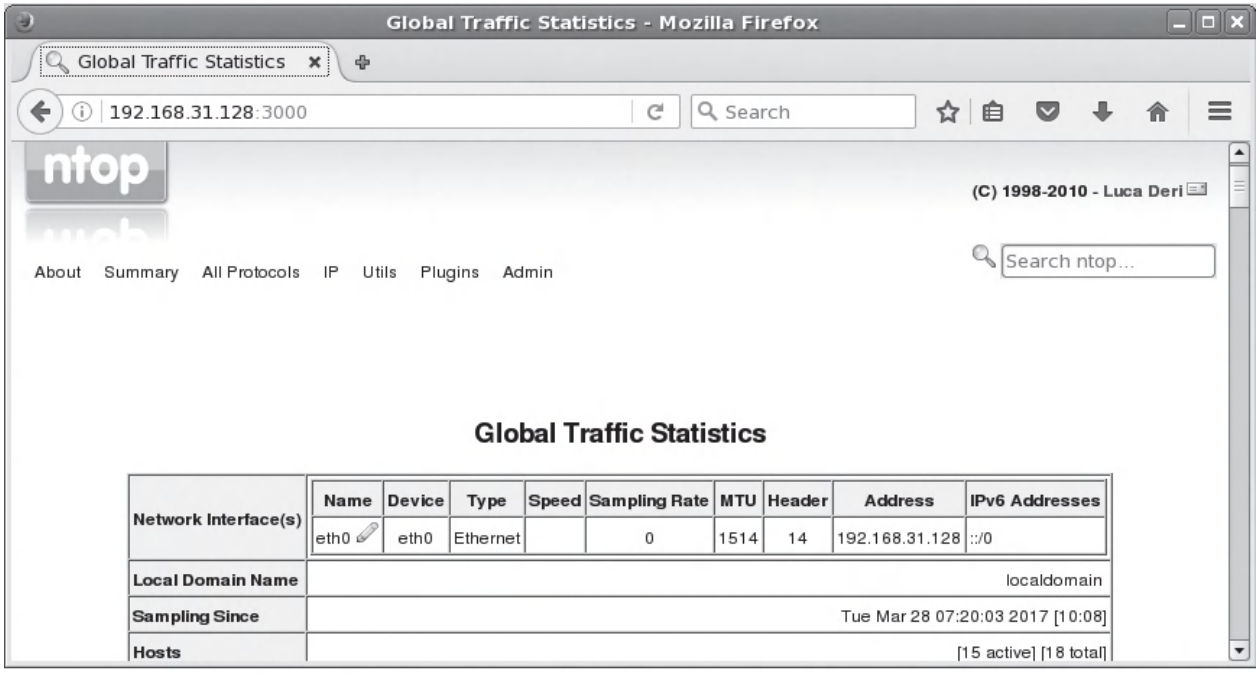


图 2-2　Ntop 的 Web 界面

知识拓展

Ntop 监控使用手册 Web 界面说明

About：在线手册。
Summary：目前网络的整体概况。
Traffic：流量。
Hosts：所有主机的使用概况。
Network Load：各时段的网络负载。
Netflows：网络流量图。
IP：各主机的流量状况与排名明细。
Traffic：所有主机的流量明细，按应用层协议分类查看各主机流量统计信息。
Multicast：多点传送情况。
Domain：域名。
Distribution：通信量状况。
Local >>Local：本地流量。
Local>>Remote：所有主机对外的明细。
Remote>>Local：远程主机到本地流量。
Remote>>Remote：远程主机到远程主机流量。
All Protocols：查看各主机占用的频宽与各时段网络使用者等的明细。
Traffic：流量。
Throughput（吞吐量）：频宽使用明细表（点选主机，可以看到该主机详细的信息及使用状况）。
Activity：各时段所有主机使用流量状况（点选主机，可以看到该主机详细的信息及使用状况）。
utils：数据备份和日志明细。
plugin：插件控制。
Admin：新增 Ntop 使用者或重新启动，停止 Ntop 服务。

2. 安装 Ntopng

Ntopng 是目前 Ntop 官方的主推版本，它以 Ntop 为基础，是下一代的 Ntop。Ntopng 是一个基于网页的高速通信分析器和流量收集器，它运行于所有 Unix 平台、Mac OS X 和 Windows。

下面简单介绍一下该软件的 yum 安装方式。

（1）配置默认 yum 源。设置 yum 源有很多种方式，登录 http://packages.ntop.org/centos/，查看页面最下方 ntop.repo 配置链接。为 Ntopng 手动创建 yum 源软件仓库。

```
[root@localhost ~]# cd /etc/yum.repos.d
[root@localhost yum.repos.d]# mv CentOS-Base.repo  CentOS-Base.repo.bak
// 备份原 repo
```

（2）安装 epel 源。安装一个“epel-release”软件包，这个软件包是企业版 Linux 附加软件包，里面包含很多基本源里没有的软件，它会完成自动配置 yum 软件仓库。

提示

如果 redis 无法安装，可下载源码包独立安装。

笔记

```
    // 网络下载 epel-release 安装包到本地
    [root@localhost ~]# wget https://mirror.lzu.edu.cn/epel/epel-release-latest-5.noarch.rpm
    // 执行 epel-release 软件包的安装
```

```
    [root@localhost ~]# rpm -ivh epel-release-latest-5.noarch.rpm
    [root@localhost ~]# yum clean all //清除 yum 下载包的缓存，headers 和 packages
    [root@localhost ~]# yum makecache //生成缓存
    [root@localhost ~]# yum -y update //更新系统
    // 安装 ntopng
    // 安装开发工具包
    [root@localhost ~]# yum -y groupinstall "Development Tools"
    // 安装依赖软件环境
    [root@localhost ~]# yum -y install tcl //安装 tcl
    [root@localhost ~]#yum -y install libpcap-devel glib2-devel GeoIP-devel libxml2-devel libcurl-devel redis autoconf automake sqlite-devel
```

```
    [root@localhost ~]# wget http://download.redis.io/releases/redis-3.2.8.tar.gz
    [root@localhost ~]# tar zxfv redis-3.2.8.tar.gz
    [root@localhost ~]# ls
    [root@localhost ~]# cd redis-3.2.8
    [root@localhost redis-3.2.8]# make
    [root@localhost redis-3.2.8]# make test
    [root@localhost redis-3.2.8]# make install
    [root@localhost redis-3.2.8]#cd src
    [root@localhost src]#ls
    [root@localhost src]#./redis-server
```

提示

如果 Ntopng 无法安装，可下载源码包独立安装，登录网页 https://sourceforge.net/projects/ntop/files/latest/download?source=files 下载安装包。

启动 redis 服务检测 redis 运行效果，如图 2-3 所示。

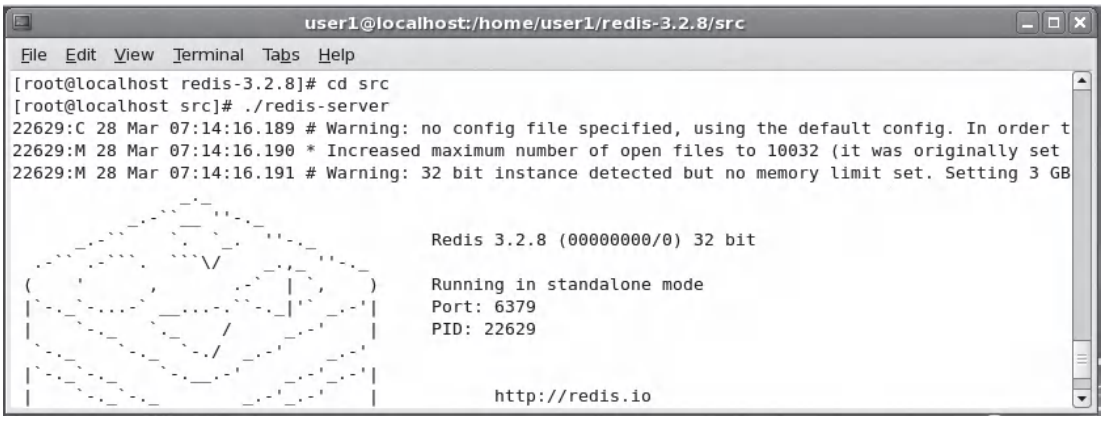


图 2-3 redis 运行效果

```
[root@localhost ~]# cd  ..
[root@localhost ~]# tar -zxvf  ntopng-2.4-stable.tar.gz
[root@localhost ~]# ls
[root@localhost ~]# cd ntopng-2.4-stable
[root@localhost ntopng-2.4-stable]# ./autogen.sh  --prefix=/usr/local/ntopng
[root@localhost ntopng-2.4-stable]# ./configure --prefix=/usr/local/ntopng
[root@localhost ~]# tar -zxvf curl-7.53.1.tar.gz
[root@localhost ~]# ls
[root@localhost ~]# cd curl-*
[root@localhost curl-7.53.1]#./configure
[root@localhost curl-7.53.1]#make & make install
// 重新执行 ntopng 的安装，不同操作系统平台下根据提示安装缺失文件后继续 ntopng 的安装
[root@localhost curl-7.53.1]#cd  ../ntopng-2.4-stable
[root@localhost ntopng-2.4-stable]# ./configure --prefix=/usr/local/ntopng
// 如果出现 MySQL libraries not found ****
[root@localhost ntopng-2.4-stable]# yum list \*mysql\* | grep dev
// 把出现的 mysql 程序都安装上
[root@localhost ntopng-2.4-stable]#]# yum  -y install ***
// 再次重新执行 ntopng 的安装
[root@localhost ntopng-2.4-stable]# ./configure --prefix=/usr/local/ntopng
You are now ready to compile typing /usr/bin/gmake
Please do not forget to download GeoIP databases doing: /usr/bin/gmake geoip
[root@localhost ntopng-2.4-stable]# /usr/bin/gmake geoip
[root@localhost ntopng-2.4-stable]# make & make install
```

（3）配置 Ntopng。Ntopng 安装完成后，yum 安装后默认主配置文件模板是 /etc/ntopng/ntopng.conf.sample，独立安装后该文件在 /usr/local/ntopng 下，可以将此文件重命名为 ntopng.conf，然后在这个配置文件中添加一些配置信息。

```
[root@localhost ~]# cd  /usr/local/ntopng  // 进入 ntopng 默认安装路径
[root@localhost ntopng]# ls  // 查看目录下包含的文件
[root@localhost ntopng]# mv  ntopng.conf.sample  ntopng.conf
[root@localhost ntopng]# vi ntopng.conf  // 编辑主配置文件
-G=/var/tmp/ntopng.gid
--local-networks "192.168.31.0/24"
--interface eth0
--user nobody
--http-port 3000
```

笔记

提示

如果仍缺少 libcurl(-dev)，可访问 http://curl.haxx.se 下载。

笔记

相关参数及其含义如下：

-G：指定存储 Ntopng 进程号的文件路径。

--local-network：指定要监控的本地子网段。

--interface eth0：指定监听 eth0 网卡上的流量。

--user：指定运行 Ntopng 服务所使用的账户。

--http-port：指定 Ntopng 的 Web 服务端口号，默认端口为 3000。

（4）启动 Ntopng 服务。启动 Ntopng 服务之前，需要先启动 redis 服务，redis 是一个 key-value 存储系统，为 Ntopng 提供键值存储。

```
[root@localhost ~]# /redis-3.2.8/src/redis-server
[root@localhost ~]# /usr/local/ntopng/bin/ntopng
[root@localhost ~]# ifconfig  // 检测服务器 IP 地址
```

打开浏览器，可以通过访问 http://yourservername:3000（http://192.168.31.128:3000）来测试 Ntopng 应用，将会看到 Ntopng 登录页面。默认登录用户名和密码均为 admin，可在登录后进行修改。

任务 2.3 网络性能评估工具 iperf

网络性能评估主要是监测网络带宽的使用率，网络设计方面的不合理、网络本身存在安全漏洞等原因，都会导致网络带宽利用率降低。定位网络带宽利用率低的原因，就需要用到一些网络性能评估工具。

子任务 2.3.1 初识 iperf

iperf 命令是一个网络性能测试工具，可以用它来测试 TCP 和 UDP 带宽质量，还可以提供网络延迟抖动、数据包丢失率、最大传输单元等统计信息。运维人员可以根据信息判断网络性能，从而定位网络瓶颈，解决网络故障。利用 iperf 这一特性，可以用来测试一些网络设备如路由器、防火墙、交换机等的性能。

iperf 分为两种版本，Unix/Linux 版和 Windows 版，Unix/Linux 版更新比较快。Windows 版的 iperf 叫 jperf 或者 xjperf。主要功能包括 TCP 和 UDP 两方面。

1）TCP 方面

（1）测试网络带宽。

（2）支持多线程，在客户端与服务器端支持多重连接。

（3）报告 MSS/MTU 值的大小。

（4）支持 TCP 窗口值自定义并可通过套接字缓冲。

2）UDP 方面

（1）设置指定带宽的 UDP 数据流。

（2）测试网络抖动值、丢包数。

（3）支持多播测试。

（4）支持多线程，在客户端与服务器端支持多重连接。

子任务 2.3.2　安装与使用 iperf

笔记

iperf 可以运行在任何 IP 网络上，包括本地以太网、接入因特网、wifi 网络等。在工作模式上，iperf 运行于 C/S 模式下，其服务器端主要用于监听到达的测试请求，而客户端主要用于发起连接会话，因此要使用 iperf 需要两台计算机，一台运行在服务器模式下，另一台运行在客户端模式下。

```
[root@localhost ~]# tar -zxvf iperf-3.0.tar.gz
[root@localhost ~]# cd iperf
[root@localhost iperf]#./configure --prefix=/usr/local/iperf
[root@localhost iperf]# make & make install
[root@localhost iperf]# cd /usr/local/iperf
[root@localhost iperf]#ls
[root@localhost iperf]#cd bin
[root@localhost bin]#./iperf -h
```

1）服务器端选项

-s：以服务器端模式启动

-D：作为后台守护进程运行。

2）客户端选项

-c：将 iperf 以客户端模式启动。

-u：指定 UDP 协议。

-b：指定带宽。

-t：指定传输总时间。

-n：指定传输字节数。

-P：指定服务器端和客户端之间使用的线程数。

-R：切换数据发送、接收模式。

-w：指定套接字缓冲区大小。

-B：绑定主机地址或端口。

-M：设置 TCP 最大信息段值。

3）客户端与服务器端公用选项

-f：指定带宽输出单位。

-p：指定服务器端或客户端端口。

-i：指定每次报告之间的时间间隔。

-F：指定文件作为数据流进行带宽测试。

子任务 2.3.3　iperf 应用实例

Iperf 使用环境需要两台计算机：服务器（192.168.31.128）和客户机（192.168.31.129），下面通过实验来体验 iperf 的使用。带宽测试通常采用 UDP 模式，因为能测出极限带宽、延时抖动、丢包率。在进行测试时，先用指定带宽进行测试，然后根据测试结果，再以实

笔记

际带宽作为数据发送速率进行测试，会发现延时抖动和丢包率有所变化，重复测试几次，得出稳定的实际带宽。

1. UDP 模式

服务器端：

```
[root@localhost ~]# /usr/local/iperf/bin/iperf -u -s
```

客户端：

```
[root@localhost ~]# /usr/local/iperf/bin/iperf -u -c 192.168.31.128 -b 100M
-t 60
//UDP 模式下，以 100 Mbps 为数据发送速率，客户端到服务器 192.168.31.128 上
传带宽测试，测试时间为 60 s。
[root@localhost ~]#/usr/local/iperf/bin/iperf -u -c 192.168.31.128 -b 5M -P
30 -t 60
// 客户端同时向服务器端发起 30 个连接线程，以 5 Mbps 为数据发送速率。
[root@localhost ~]#/usr/local/iperf/bin/iperf -u -c 192.168.31.128 -b 100M
-d -t 60
// 以 100 M 为数据发送速率，进行上下行带宽测试。
```

2. TCP 模式

服务器端：

```
[root@localhost ~]#/usr/local/iperf/bin/iperf -s
```

客户端：

```
[root@localhost ~]#/usr/local/iperf/bin/iperf -c 192.168.31.128 -t 60
//TCP 模式下，客户端到服务器 192.168.31.128 上传带宽测试，测试时间为 60 s。
[root@localhost ~]#/usr/local/iperf/bin/iperf -c 192.168.31.128  -P 30 -t 60
// 客户端同时向服务器端发起 30 个连接线程。
[root@localhost ~]#/usr/local/iperf/bin/iperf -c 192.168.31.128  -d -t 60
// 进行上下行带宽测试。
```

任务 2.4　网络探测和安全审核工具 nmap

Linxu 服务系统提供了一个网络探索工具和安全扫描器——nmap，它是一种自动检测远程或本地主机安全弱点的程序。使用 nmap 可以获得远程服务器的大量信息，通过对这些信息的分析，进一步了解远程主机存在的安全问题。

子任务 2.4.1　了解 nmap 和 Zenmap

nmap 是一个端口扫描工具，它提供了很多扫描技术。nmap 可以扫描一台主机，只

需指定主机名或 IP 地址即可；也可以扫描一个网段，如 192.168.1.1/24、192.168.1.*、192.168.1.0~254。用户可以选择 namp 工具扫描的网络类型，也可以选择扫描特定的一个或多个端口，如 80、20~30。

笔记

nmap 是 Network Mapper 的缩写，由 Fyodor 在 1997 年创建，现在已经成为网络安全必备的工具之一，更多详细信息可以参考官方网站：http://www.nmap.org。nmap 作为一个流行的安全工具，它的主要特点如下：

（1）非常灵活。nmap 支持 10 种以上的扫描方式，并支持多种目标对象扫描。

（2）支持主流操作系统 Windows、Linux、BSD、Solaris、AIX、Mac OS 等多种平台，可移植性强。

（3）使用简单。nmap 安装、使用都非常简单。

（4）自由软件。nmap 是在 GPL 协议下发布的，在 GPL License 的范围内可自由使用。

Zenmap 是 nmap 的 GUI 版本，由 nmap 官方提供，通常随着 nmap 安装包一起发布。Zenmap 是用 Python 语言编写的，能够在 Windows、Linux、UNIX、Mac OS 等不同系统上运行。开发 Zenmap 的目的主要是为 nmap 提供更加简单的操作方式。

子任务 2.4.2 nmap 基本功能与结构

1. nmap 的功能

从它实现功能的方向性来划分，主要有如下功能：

（1）主机探测。它能够探测网络上的各个主机。

（2）端口扫描。探测目标主机所开放的端口情况。

（3）版本检测。探测目标主机提供的网络服务名称及版本。

（4）系统检测。探测目标主机操作系统及网络设备。

2. nmap 的结构

nmap 源码包解压后，可以看到目录下有许多的子目录和文件。

1）从文件组织方式上

（1）nmap 核心功能源码。

（2）nmap 核心数据库文件。

（3）编译链接相关文件。

（4）其他。

2）从文件类型上

（1）C/C++ 文件。主要实现 nmap 核心功能（主机探测、端口扫描、服务侦测等）。

（2）Python 文件。主要实现图形界面。

（3）Lua 和 NSE 文件。提供扫描脚本。

（4）XML 文件。

（5）其他。

nmap 利用上述几个功能探测网络上的主机和服务，发送特制的数据包，在发现目标的主机后，接着进行端口扫描，进而通过扫描到的端口确定运行的应用程序类型以及版本信息，对返回的数据包进行分析，最终确定操作系统的版本及漏洞信息，并绘制网络拓扑图。另外，nmap 还提供了防火墙与入侵检测系统的规避技巧，这个功能可以应用于基本

笔记

功能的各个阶段。最后，nmap 还通过 NSE（Nmap Scripting Language）脚本引擎功能对自身基本功能进行补充和扩展。

子任务 2.4.3 安装与验证 nmap

nmap 的安装非常简单，官方提供源码编译安装和 RPM 包两种方式，可根据自己的情况选择安装不同方式进行安装。

1. 源码编译安装

从官方网站下载源码包，然后编译安装即可。

```
[root@localhost ~]# wget https://nmap.org/dist/nmap-7.40.tar.bz2
[root@localhost ~]# tar -jxvf nmap-7.40.tar.bz2
[root@localhost ~]#ls
[root@localhost ~]# cd nmap-7.40
[root@localhost nmap-7.40]#./configure --prefix=/usr/local/nmap
[root@localhost nmap-7.40]# make  & make install
[root@localhost nmap-7.40]# cd /usr/local/nmap
[root@localhost nmap]# ls
[root@localhost nmap]# cd bin
[root@localhost nmap]# ./nmap -h  // 查看帮助信息确认安装成功
```

至此，源码编译方式安装 nmap 完成。

2. RPM 包安装

nmap 官方也提供 RPM 格式的安装包，直接从网站下载 RPM 格式的安装包，然后进行安装即可，操作过程如下：

```
[root@localhost ~]# wget http://nmap.org/dist/nmap-6.40-1.x86_64.rpm
[root@localhost ~]# rpm -ivh nmap-6.40-1.x86_64.rpm
[root@localhost ~]# namp -h  // 查看安装成功后的帮助信息
```

子任务 2.4.4 nmap 的典型用法

nmap 命令语法相当简单，如果以 root 身份来运行 nmap，那它的功能会大大增强。

1. 语法与参数说明

语法格式：namp [扫描类型][选项] 目标主机或网络

参数及其含义：

1）扫描类型

-sT：TCP 链接扫描，这是对 TCP 最基本的侦测形式。

-sS：TCP SYN 扫描。

-sF、-sX、-sN：Xmas Tree 或 Null 扫描模式。

-sP：ping 扫描。

笔记

-sU：UDP 扫描。
-s0：IP 协议扫描。
-sI：高级扫描，允许对隐藏 TCP 端口进行扫描。
-sA：ACK 扫描。
-sW：windows 扫描。
-sR：RPC 扫描。
-sL：列表扫描。
2）一般选项
-P0：扫描前不 ping 主机。
-PT：用 TCP 的 ping 来确定主机是否打开。
-PS：用 SYN 包代替 ACK 包。
-PU：向指定的主机发送 UDP 探测。
-6：启用 ipv6 支持。
-v：详细模式。
-h：显示帮助信息。
3）时间选项
-T：表达时间优先权的参数设置。
--host_timeout：指定扫描时间总量，超时放弃。
--max_rtt_timeout：指定从远程目标返回回应的最大时间，默认 9 000 ms。
--min_rtt_timeout：指定探测往返时间。
--initial_rtt_timeout：指定最初探测器的 timeout 时间。
--max_parallelism：指定 nmap 允许的最大并行扫描数目。
--min_parallelism：指定 nmap 允许的最少并行扫描数目。
--scan_delay：指定两个探测之间的最小时间间隔。

```
[root@localhost ~]# /usr/local/nmap/bin/nmap  192.168.31.128
```

nmap 监控结果如图 2-4 所示。

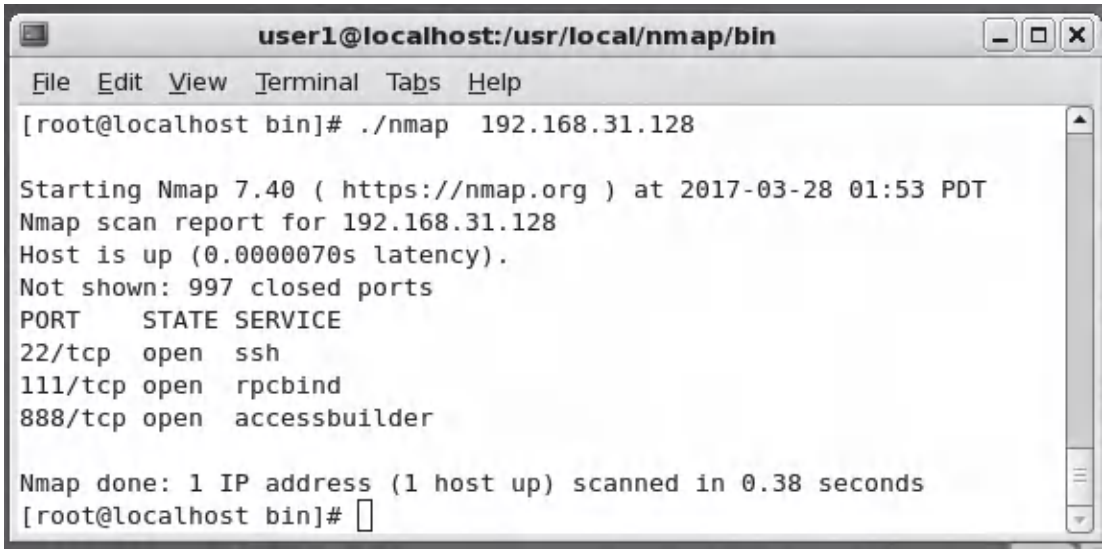


图 2-4　nmap 监控结果

笔记

由图可知，目标主机“192.168.31.128”处于“up”状态，并且此主机上开放了 22、111、888 端口，同时还侦测到每个端口对应的服务。

```
[root@localhost ~]#nmap -T4 -A -v 192.168.31.128
```

其中参数及其含义如下：

-A：用于开启全面扫描。

-T4：指定扫描过程中使用的时序模板。

-v：显示扫描细节。

2. 扫描结果

第一部分：对主机是否在线进行扫描；

```
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 02:55
Completed Parallel DNS resolution of 1 host. at 02:55, 0.03s elapsed
Initiating SYN Stealth Scan at 02:55
```

第二部分：对端口进行扫描；

```
Scanning 192.168.31.128 [1000 ports]
Discovered open port 22/tcp on 192.168.31.128
Discovered open port 111/tcp on 192.168.31.128
Discovered open port 888/tcp on 192.168.31.128
Completed SYN Stealth Scan at 02:55, 0.08s elapsed (1000 total ports)
Initiating Service scan at 02:55
```

第三部分：对运行的服务进行扫描；

```
Scanning 3 services on 192.168.31.128
Completed Service scan at 02:55, 14.65s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 192.168.31.128
NSE: Script scanning 192.168.31.128.
Initiating NSE at 02:55
Completed NSE at 02:55, 0.08s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Nmap scan report for 192.168.31.128
Host is up (0.0014s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE VERSION
```

```
22/tcp   open   ssh   OpenSSH 4.3 (protocol 2.0)
111/tcp open   rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version   port/proto   service
|   100000   2   111/tcp   rpcbind
|   100000   2   111/udp   rpcbind
|   100024   1   885/udp   status
|_  100024   1   888/tcp   status
888/tcp open   status   1 (RPC #100024)
```

第四部分：对操作系统类型和版本进行探测；

```
Device type: general purpose
Running: linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.18
OS details: linux 2.6.18
Uptime guess: 0.386 days (since Mon Mar 27 17:39:59 2017)
```

第五部分：对目标主机的路由跟踪信息。

```
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=203 (Good luck!)
IP ID Sequence Generation: All zeros
NSE: Script Post-scanning.
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Initiating NSE at 02:55
Completed NSE at 02:55, 0.00s elapsed
Read data files from: /usr/local/nmap/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.31 seconds
Raw packets sent: 1019 (45.598KB) | Rcvd: 2044 (87.032KB)
```

子任务 2.4.5　nmap 主机发现扫描

主机发现扫描主要用来判断目标主机是否在线，先要发送探测数据包到目标主机，如果能收到回复，那么认为目标主机处于在线状态。nmap 支持多种不同的主机探测方法，用户可在不同的环境下选择不同的方式来探测目标主机。具体可通过以下选项来定制主机发现的探测方式。

笔记

笔记

-sn：进行主机扫描。

-Pn：进行端口扫描。

-sL：显示目标主机 ip 列表。

-PS/PZ/PU/PY：指定 nmap 使用 TCP SYN、TCP ACK、UDP、SCTP 方式进行主机扫描。

-PE/PP/PM：指定 nmap 使用 ICMP echo、timestamp、netmask 请求报文方式扫描主机。

-PO：使用 IP 协议包探测目标主机。

-n/-R：指定是否使用 DNS 解析。

```
[root@localhost ~]# nmap  -sn -PE -PS22,80 -PU53 192.168.31.12
```

在这个例子中，使用了“-PE”“-PS”“-PU”等参数，根据上面的介绍，“-PE”是以发送 ICMP echo 报文的形式进行主机探测的，“-PS”是以发送 TCP SYN/ACK 包的形式侦测主机信息的，而“-PU”则是以 UDP 的方式进行主机侦测的。

子任务 2.4.6 nmap 端口扫描

端口扫描是 nmap 最核心的功能，通过端口扫描可以发现目标主机上 TCP、UDP 端口的开放情况。nmap 在默认状态下会扫描 1 000 个最有可能开放的端口，并将侦测到的端口状态分为 6 类：

Open：表示端口是开放的。

Closed：表示端口是关闭的。

Filtered：表示端口被防火墙屏蔽。

Unfiltered：表示端口没有被屏蔽。

open|filtered：表示不确定状态。

closed|filtered：表示不确定状态。

在端口扫描方式上，nmap 支持 10 种以上的探测方法，最常用的有“TCP SYN scanning”，这是默认的端口扫描方式，另外还有“TCP connect scanning”“TCP ACK scanning”“TCP FIN/Xmasscanning”“UDP scanning”等探测方式。

扫描类型：

-sT：TCP 链接扫描，这是对 TCP 最基本的侦测形式。

-sS：TCP SYN 扫描。

-sF、-sX、-sN：Xmas Tree 或 Null 扫描模式。

-sA：ACK 扫描。

-sW：windows 扫描。

-p<port ranges>：指定端口扫描。

-F：快速扫描。

--top-ports n：仅扫描开放率最高的 n 个端口。

```
[root@localhost ~]# nmap -sU -sS -F www.zhy.com
```

参数“-sS”表示使用 TCP SYN 方式扫描 TCP 端口，“-sU”表示扫描 UDP 端口，“-F”表示使用快速扫描模式，扫描最可能开放的前 100 个端口（TCP 和 UDP 各 100 个端口）。

子任务 2.4.7　nmap 版本侦测

笔记

nmap 的版本侦测功能主要用来确定目标主机开放的端口上运行的应用程序及版本信息，nmap 的版本侦测支持 TCP/UDP 协议，支持多种平台的服务侦测，支持 IPV6 功能，并能识别几千种服务签名。

-sV：设置 nmap 进行版本侦测。

-version-intensity n：设置强度值，取值范围 0~9，值越高，越精确。

-version-light：设置轻量级检测方式。

-version-all：设置最高强度值检测。

-version-trace：显示详细过程。

```
[root@localhost ~]# nmap  -sV  172.22.55.21
```

通过对服务器上运行服务的了解，以及对服务版本的探测，判断服务器是否存在软件漏洞，进而提醒运维管理人员进行端口关闭或升级软件等操作，尽早应对可能出现的安全威胁。

子任务 2.4.8　nmap 操作系统侦测

操作系统侦测主要是对目标主机运行的操作系统类型及版本信息进行检测。nmap 拥有丰富的系统库，可以识别近 3 000 种操作系统与设备类型。

-O：设置 nmap 进行操作系统侦测。

--osscan-guess：猜测目标主机的操作系统类型。

```
[root@localhost ~]# nmap -O --osscan-guess 192.168.31.128
```

nmap 扫描主机输出结果如图 2-5 所示。

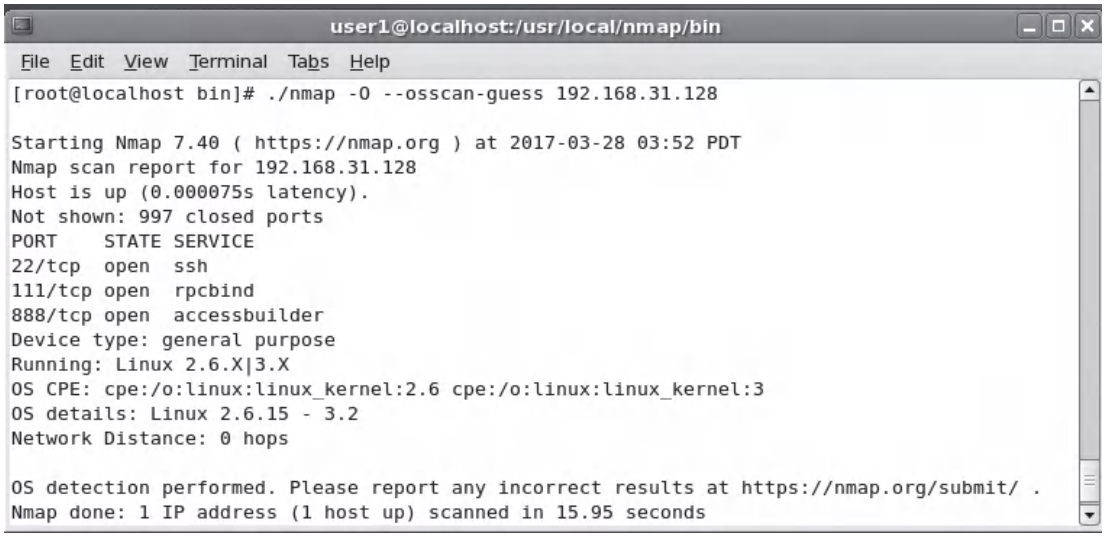


图 2-5　nmap 扫描主机输出结果

笔记

指定“-O”选项后，nmap 命令首先执行主机发现操作，接着执行了端口扫描操作，然后根据端口扫描的结果进行操作系统类型的侦测，获取到的信息有设备类型、操作系统版本、操作系统的 CPE 描述、操作系统的细节和网络距离。如果不能确定操作系统的版本，会猜测每个系统版本的可能性比率，例如对于“192.168.31.128”主机，nmap 给出最可能的操作系统版本是 Linux 2.6.15-3.2。