# centos tar压缩与解压缩命令大全

管理centos服务器的时候常会对文件进行一些常规操作，除了ftp之外了解在ssh下必要的文件操作命令那也是必不可少的，以下摘录一些常用的文件操作命令：

**文件操作：**

ls      ####查看目录中的文件####

ls -F       ####查看目录中的文件####

ls -l       ####显示文件和目录的详细资料####

ls -a       ####显示隐藏文件####

ls \*[0-9]\*       ####显示包含数字的文件名和目录名####

rm go.tar       #### 删除go.tar文件####

find mt.cgi       ####查找文件名为mt.cgi的文件####

df ?h       ####查看磁盘剩余空间####

find / -name xxx -print       ####查找xxx文件####

**压缩与解压缩：**

tar xvf wordpress.tar       ####解压tar格式的文件####

tar -tvf myfile.tar       ####查看tar文件中包含的文件 ####

tar cf toole.tar tool       ####把tool目录打包为toole.tar文件####

tar cfz xwyme.tar.gz tool      ####把tool目录打包且压缩为xwyme.tar.gz文件，因为.tar文件几乎是没有压缩过的，MT的.tar.gz文件解压成.tar文件后差不多是10MB ####

tar jcvf /var/bak/www.tar.bz2 /var/www/       ####创建.tar.bz2文件，压缩率高####

tar xjf www.tar.bz2       ####解压tar.bz2格式####

gzip -d ge.tar.gz       ####解压.tar.gz文件为.tar文件####

unzip phpbb.zip       ####解压zip文件，windows下要压缩出一个.tar.gz格式的文件还是有点麻烦的####

bunzip2 file1.bz2       ####解压一个叫做 ‘file1.bz2′的文件####

bzip2 file1       ####压缩一个叫做 ‘file1′ 的文件####

gunzip file1.gz       ####解压一个叫做 ‘file1.gz’的文件####

gzip file1       ####压缩一个叫做 ‘file1′的文件####

gzip -9 file1       ####最大程度压缩####

rar a file1.rar test\_file       ####创建一个叫做 ‘file1.rar’ 的包####

rar a file1.rar file1 file2 dir1       ####同时压缩 ‘file1′, ‘file2′ 以及目录 ‘dir1′####

rar x file1.rar       ####解压rar包####

unrar x file1.rar       ####解压rar包####

tar -cvf archive.tar file1       ####创建一个非压缩的 tarball####

tar -cvf archive.tar file1 file2 dir1       ####创建一个包含了 ‘file1′, ‘file2′ 以及 ‘dir1′的档案文件####

tar -tf archive.tar       ####显示一个包中的内容####

tar -xvf archive.tar       ####释放一个包####

tar -xvf archive.tar -C /tmp       ####将压缩包释放到 /tmp目录下####

tar -cvfj archive.tar.bz2 dir1       ####创建一个bzip2格式的压缩包####

tar -xvfj archive.tar.bz2       ####解压一个bzip2格式的压缩包####

tar -cvfz archive.tar.gz dir1       ####创建一个gzip格式的压缩包####

tar -xvfz archive.tar.gz       ####解压一个gzip格式的压缩包####

zip file1.zip file1       ####创建一个zip格式的压缩包####

zip -r file1.zip file1 file2 dir1       ####将几个文件和目录同时压缩成一个zip格式的压缩包####

unzip file1.zip       ####解压一个zip格式压缩包####

# 每个极客都应该知道的Linux技巧

是什么区分开普通的Linux用户和超级极客的呢？很简单：用在学习那些使长时工作转变为一瞬间工作的技巧、诀窍、秘诀和技术上的时间。如果你想提高效率而又不用做这些搜集资料的跑腿活儿，那我们已经收集了50多条简单易学的Linux技巧，以帮助你更潇洒地工作，并且最大限度的利用你的电脑。开始享受吧！

**1：检查不是由你运行的程序**

难度：高级

应用程序：bash（译注：UNIX或者LINUX的shell）

想象下这个场景-你已经准备好了要和同事在办公室里对战一局快速的Crack Attack（译注：一款免费的OpenGL游戏，类似俄罗斯方块），不料却发现，当你正准备要击败你这个傲慢的下属时，方块却停止了-是什么让你的机器变得这么慢？这肯定是因为其他的用户运用他们娴熟的经验来盗取你宝贵的CPU时间、服务器或者其他的诡异方式，真烦人！

好，让我们列举出系统中所有不是由你运行的程序！

1. ps aux | grep -v `whoami`

或者，更巧妙些的是，为什么不列出最占用时间的前十个程序呢：

1. ps aux--sort=-%cpu | grep -m 11 -v `whoami`

也许最好使用root权限来运行这条指令，因为这样可以过滤出大部分活跃的后台程序。现在你有了这些信息，你可以终止他们的进程，但是更“卑鄙”的是反复地在他们的桌面上运行xeyes（译注：Linux下的一条好玩的命令）！

**2：在多个文件中替换掉相同的文本**

难度：中等

应用程序：find（译注：Linux下的一条命令）/Perl

如果你有个文件，想在多个位置进行替换，这里有很多方法来实现。调用test[someting]把当前目录里所有文件中的Windows替换成Linux，你可以像这样运行它：

1. perl -i -pe 's/Windows/Linux/;' test\*

要替换当前目录以及下层目录里所有文件中的Windows为Linux，你可以这样运行：

1. find . -name '\*.txt' -print | xargs perl -pi -e's/Windows/Linux/ig' \*.txt

或者如果你更需要让它只作用于普通文件上：

1. find -type f -name '\*.txt' -print0 | xargs --null perl -pi -e 's/Windows/Linux/'

节省了大量的时间并且获得了高水平的大师等级！

**3：合并一个不稳定的终端**

难度：简单

应用程序：bash

我们都已经做过这件事了-偶尔使用less或者cat来列出一个文件，或者以浏览二进制文件而结束，这通常包含各种可以轻易弄乱你的终端显示的控制代码、警报声、一些有趣的字符，以及一些奇怪的颜色组合，最后，你的字体将会被一些难以辨认的字符所替代，而你却不知道该怎么办。显然bash仍然继续工作着，但是你就是不能明白到底发生了什么！发送一个初始化命令给这个终端：

1. reset

然后一切就再次恢复正常了。

**4：创造Mozilla关键词**

难度：简单

应用程序：Firefox/Mozilla

在浏览器上的一个有用的特征是，它们有一种能力，可以通过输入gg onion来通过Google搜索onion这个词的一切。同一功能在Mozilla上也可实现，通过单击Bookmarks（书签）>Manage Bookmarks（管理书签）然后添加一个新的书签，添加的URL就像这样：

1. http://www.google.com/search?q=%s

现在选择书签编辑器中的条目并且点击Properties（属性）按钮，现在输入一个类似gg的关键字（或者可以是你选择的任何字符）就行了。在URL中的%s将被关键字之后的文本内容替代。你可以用这种方法向其他一些依赖你当前URL信息的网站发送请求。

或者，右击一个搜索区域然后选择菜单选项“为这个搜索添加一个关键词…”。接下来的会话将允许你使用一个指定的关键词。

**5：运行多种X会话**

难度：简单

应用程序：X

如果你给别人分享了你的Linux box（），而厌烦了不断的登入、登出，当你知道了这不是必要的时候，可能会如释重负。假设你的电脑以图形模式启动（runlevel 5），通过同时按下Control+Alt+F1键-你将得到一个登陆提示。输入你的登录名以及密码然后执行：

1. startx -- :1

来进入你的图形环境。按下Ctrl+Alt+F7来回到在你之前的用户会话，如果想回到你自己的用户会话则按下Ctrl+Alt+F8。

你可以重复使用这项技巧：F1键到F6键可以识别六个控制台会话，而F7到F12可以识别六个X会话。警告：尽管这在多数情况下是适用的，但是不同的变种可能用不同的方式来实现这项功能。

**6：更快地浏览**

难度：简单

应用程序：KDE（译注：一种Linux的桌面环境）

KDE是一个虽然小众，但是在提升你的网页浏览经验方面很有用的选择。启动一个KDE控制中心，然后从工具条中选择System（系统）>KDE performance。现在你可以选择预加载浏览器实例。实际上，这意味著浏览器在开机的时候就开始运行，在你用之前都保持隐藏状态。当你想要使用它的时候，它几乎是瞬间出现。

**7：简单地备份你的网站**

难度：简单

应用程序：Backups

如果你想要从一台电脑上备份一个目录，但你仅仅想要复制改变的文件到它的备份而不是复制所有的东西到各自的备份，你可以使用工具rsync来实现它。你需要在这个远程的源备份计算机上有一个账户。下面是这条命令：

1. rsync -vare ssh jono@192.168.0.2:/home/jono/importantfiles/\* /home/jono/backup/

这样我们就备份了192.168.0.2地址上/home/jono/importantfiles/目录下的所有文件到当前机器上的/home/jono/backup目录下。

**8：使你的时钟保持准时**

难度：简单

应用程序：NTP

如果你发现电脑上的时钟似乎与时间脱轨了，你可以使用一个特殊的NTP工具，来确保它准到只有穿实验服的人才会需要的程度。你需要安装ntpdate工具，它通常包含在NTP包中，然后你就可以与NTP服务器保持同步了：

1. ntpdate ntp.blueyonder.co.uk

可以在www.eecis.udel.edu/~mills/ntp/clock1b.html上获得相应的NTP服务器的列表。如果你修改了你的开机程序并且在脚本中包含这些命令，就可以确保无论何时你启动电脑都是完全准时的。你还可以运行一个定时任务来校对时间。

**9：找到最大的文件**

难度：简单

应用程序：Shell

计算机中一个常见的问题是，你可能想删一部分大的文件（类似音频或视频片段）。如下你可以找到当前目录中最大的那个文件：

1. ls -lSrh

“r”的作用是将大的文件列在后面，而“h”则是给出易于人们阅读的输出(MB或者诸如此类)。你也可以搜寻最大的MP3/MPEG文件：

1. ls -lSrh \*.mp\*

你也可以通过下面这条命令搜寻最大的目录：

1. du -kx | egrep -v "\./.+/" | sort -n

**10：Nautilus快捷键**

难度：简单

应用程序：Nautilus（译注：Linux图形界面中的一个文件管理器）

尽管很多文件管理器近日来被设计为使用鼠标进行管理，但有时用键盘还是方便些。Nautilus有一些键盘快捷键，可以使你快速浏览文件：

* 打开一个定位文件 – Ctrl+L
* 打开父文件夹 – Ctrl+Up
* 用方向键来浏览当前文件夹

你也可以使用‘emblems’来定制文件图标。这里有少量的应用于个别文件或者组的图象展示。打开Edit(编辑) > Backgrounds and Emblems(背景和符号)菜单条目，然后拖放你想要的图片即可。

**11：整理你的数据库**

难度：简单

应用程序：MySQL

无论何时你改变一个MySQL数据库的结构，或者从中删掉大量的数据，文件都有可能被分散，并导致一些性能损失。尤其是当运行查询命令时这个情况最容易发生。每当你要改动数据库的时候，记得运行这个优化器：

1. mysqlcheck -o **<databasename>**

也许你会发现，如果你使用VARCHAR文件，有必要定期的整理你的数据库表：这个变长列（variable-length columns）是非常倾向于分裂的。

**12：更快的邮件**

难度：简单

应用程序：KMail

连三秒钟都不想花费在找邮箱客户端上面？不想被桌面上堆积成山日渐腐烂的图标搞得找不到鼠标指针？不管你正在KDE里干什么，按几下键盘就可以发邮件了。输入：

1. mailto:plop@ploppypants.com

按下return，KMail将会自动启动，为你的至理名言做好准备。你甚至不需要填写完整的邮件地址。这对Internet地址也是有效的：尝试输入www.slashdot.org来启动浏览器。

**13：同化你的构造**

难度：简单

应用程序：GCC

如果你运行一个拥有适量RAM的多处理机系统（SMP），你会发现，在写代码时，运行一个parallel make很有好处。与按照默认方式运行make并执行serial builds相比，parallel build 是一种很有效的改善方法。让make在building的时候允许同时存在多个child，用如下的-j转换即可：

1. make -j4; make -j4 modules

14：节省电池电源

难度：中等

应用程序：hdparm

很可能你已经对使用hdparm来调整一个硬件驱动很熟悉了，但是它还能节约你的笔记本的电池寿命，或者通过降低硬盘转速来使生活更宁静。

1. hdparm -y /dev/hdb
2. hdparm -Y /dev/hdb
3. hdparm -S 36 /dev/hdb

按顺序，这些命令将让硬件转换成Standby（待命）模式，再转换成Sleep（睡眠）模式，并且最终设置Automatic（自动）旋转超时。这个结尾包含一个以5秒为单位的一块块数字变量（例如，值为12就等同于一分钟）。

顺便说一下，这个降低旋转时间至5秒为单位的习惯真值得拿一个特别的用户友好奖——这个值的设置也许有什么历史原因，但我们不知道。如果你恰好知道为什么，请来信告知我们！

**15：无线速度管理**

难度：中等

应用程序：iwconfig

一个radio发送/接受设备和其他设备间的传输速度，取决于有多少可用信号。在信号减弱的时候如需维持传输，就需要降低radio的传输速度。一般情况下，radio都会自己找到可用信号，并自动选择最快的速度。

在边缘区域中，信号强度勉强够用，这时候就可能因radio不断调试连接速度而造成不必要的包丢失。如果不能加强天线增益，或移动设备位置来获取更充足的信号，可以尝试强制让card同步到一个较低的速率。这样，radio就会减少调试频率，实际上，还会比使用持续触发的链接传输得快一些。不同的驱动器有不同的链接速度设置方法。在Linux中，可用iwconfig来设置速度：

1. iwconfig eth0 rate 2M

这个指令能强制使广播总是同步在2Mbps，即使还存在其他可用速度。你也可以设置一个明确的速度作为上限，允许卡自动按照任何低于这个上限的速度来工作，但是不能高于它。例如，你可以把它作用于上文举例的连接上：

1. iwconfig eth0 rate 5.5M auto

如上的指令可以让驱动器将速度上限设为5.5Mbps，只慢不快。想要将card恢复为全自动缩放，只需要制定它自己自动工作。

1. iwconfig eth0 rate auto

通常卡在1Mbps时比它们在11Mbps时可以延伸的更远。在这两种速率之间，Orinoco卡有12dB的差别，即降低数据速率可获得的潜在传输距离的四倍。

**16：疏通开端口**

难度：中等

应用程序：netstat

要获取在Linux服务器上处于监听状态的网络服务器的列表，可使用netstat程序：

1. root@catlin:~# netstat -lnp
2. Active Internet connections (only servers)
3. Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
4. tcp 0 0 0.0.0.0:5280 0.0.0.0:\* LISTEN 698/perl
5. tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 217/httpd
6. tcp 0 0 10.42.3.2:53 0.0.0.0:\* LISTEN 220/named
7. tcp 0 0 10.42.4.6:53 0.0.0.0:\* LISTEN 220/named
8. tcp 0 0 127.0.0.1:53 0.0.0.0:\* LISTEN 220/named
9. tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 200/sshd
10. udp 0 0 0.0.0.0:32768 0.0.0.0:\* 220/named
11. udp 0 0 10.42.3.2:53 0.0.0.0:\* 220/named
12. udp 0 0 10.42.4.6:53 0.0.0.0:\* 220/named
13. udp 0 0 127.0.0.1:53 0.0.0.0:\* 220/named
14. udp 0 0 0.0.0.0:67 0.0.0.0:\* 222/dhcpd
15. raw 0 0 0.0.0.0:1 0.0.0.0:\* 7 222/dhcpd

|  |
| --- |
| ​ |

如上消息告诉你PID为698的进程是一个绑定在5280端口的Perl进程。如果你不是超级用户，系统将不会显示哪个程序是运行在哪个端口上的。

**17：更快的硬件驱动**

难度：高级

应用程序：hdparm

也许你已经知道hkparm工具可以用于测试硬盘速度和更改一些设置。它还可以用于优化驱动性能，并且开启一些默认关闭的功能。在我们开始之前，敬告您，改变驱动选项可能会造成数据损坏，所以首先请备份你所有的重要数据。如下命令可以用来测试速度：

1. hdparm -Tt /dev/hda

你将会看到类似于这样的东西：

1. /dev/hda:
2. Timing buffer-cache reads: 128 MB in1.64 seconds =78.05 MB/sec
3. Timing buffered disk reads:64 MB in 18.56 seconds = 3.45MB/sec

现在我们可以尝试让它提速。找出你的驱动目前设置使用的是哪个选项，然后传递给hdparm这个硬件的名称：

1. hdparm /dev/hda
2. /dev/hda:
3. multcount=16 (on)
4. I/O support=0 (default 16-bit)
5. unmaskirq=0 (off)
6. using\_dma=0 (off)
7. keepsettings =0 (off)
8. readonly =0 (off)
9. readahead=8 (on)
10. geometry = 40395/16/63, sectors = 40718160, start = 0

这相当于一个默认设置。多数的Linux发行版本都会选择一个可以运行在多数硬件上的安全选项。为了获得更快的速度，你可能想要激活dma模式，然后当然要调整I/O支持。多数现代计算机支持模式3，这是一个近乎双倍通量的32位转换模式。你也可以尝试：

1. hdparm -c3 -d1/dev/hda

然后返回这个速度，查看一下不同。检验你的硬件支持哪些模式，然后可以在hdparm的man手册中学习如何设置它们。

**18：在你手中的可用时间**

难度：高级

应用程序：Perl

对计算来说，浪费的资源就是那些本可以派上更好的用场的资源。不管你还运行了什么别的程序，为什么不尝试运行一个进程，在终端标题栏实时显示当前的平均负载呢？

将这个存为一个脚本并命名为tl，然后把它存到你的~/bin目录下：

1. #!/usr/bin/perl -w
2. use strict;
3. $|++;
4. my $host=`/bin/hostname`;
5. chomp $host;
6. while(1) {
7. open(LOAD,"/proc/loadavg") || die "Couldn't open /proc/loadavg: $!\n";
8. my @load=split(/ /,**<LOAD>**);
9. close(LOAD);
10. print "$host: $load[0] $load[1] $load[2] at ", scalar(localtime);
11. print "07";
12. sleep 2;
13. }

|  |
| --- |
| ​ |

如果你想用用load average和你登陆的机器的当前时间，来替换titlebar名字，与西宁tl&即可。它会很恰当的在后台运行，即使你运行一个类似Vim的交互式程序。

**19：不使用X完成抓取一次屏幕截图**

难度：简单

应用程序：Shell

确实有很多屏幕捕获的工具，但是他们大部分都是基于X的。这导致了一个问题，当你运行一个X应用程序时，将会干扰这个你想要抓取的应用程序——或许是一个游戏或者甚至是一个Linux安装程序。但如果你使用了超棒的ImageMagick（译注：一个开源的图片处理工具集和开发包）导入命令，你就可以通过控制台来对X模块进行抓取。很简单，打开一个虚拟终端（例如Ctrl+Alt+F1）然后输入如下：

1. chvt 7; sleep 2; import -display :0.0 -window root sshot1.png; chvt 1;

chvt命令改变了虚拟终端，而sleep命令给重绘屏幕提供了一个时间。导入命令会在最后的chvt命令把你再次打回到虚拟终端之前，捕获所有的显示并且保存到一个文件中。请确保你把所有的命令都输入在同一行中。

这是甚至在Linux安装程序上也是有效的，很多安装程序都只留下一个控制台而转去后台运行——一个一流的到处运行的屏幕抓取器只需要加载一些它所需要的库到一个floppy（译注：磁盘）/CD上。

**20：远程访问你的程序**

难度：简单

应用程序:X

如果你想要躺在床上使用你的Linux笔记本从你的Window机器上访问你的应用程序，可以使用SSH来实现。首先你需要在/etc/ssh/sshd\_config中激活以下设置：

1. X11Forwarding yes

现在我们可以使用如下命令在192.168.0.2上运行GIMP：

1. ssh -X 192.168.0.2 gimp

**21：用到man手册**

难度：简单

应用程序：man

如果你在一个特殊的项目或者命令中需要帮助，man手册是一个好的入门。通常你使用man<command>来访问man手册，但是你也可以通过一个特殊的man页描述的关键词来搜索man页。举例来说，搜索所有详述login的man页：

1. man -k login

当你访问了一个man页，你也可以使用斜杠键来搜索一个特定的词而不是用man页本身。在你的键盘上按下“ / ”然后输入搜索项是非常简单的。

**22：和你的医生对话**

难度：简单

应用程序：Emacs

如果说Emacs只是个文本编辑器，就好像说Triumph只是个摩托车，或者说世界杯就是个四年一度的足球赛。好吧，它确实是个文本编辑器，但那只是一小~~~部分。举个栗子？打开编辑器，按下Esc键，紧接着按下X，然后进入doctor：你将会进入到一个离奇的对话中，有一个假想的但是熟练的精神治疗师。如果你想要磨蹭你的时间，这是一个很好的方法。

1. Esc-X tetris

会将你的‘编辑器’转变成一个老的、受欢迎的游乐场游戏。

疯狂到此为止了吗？绝对没有！检查你的发行版本的包，列出他们在你的Emacs上捆绑的其他功能：我们看到有国际象棋，Perl集成，IRC chat（译注：一种聊天工具），法文翻译，HTML转换，一个Java开发环境，灵巧的编辑，甚至还有一个被称为“semantic bovinator”（语义纠错）的东西。我们确实对最后一个能做什么的毫无头绪，但是我们断定无论如何你都敢尝试它！（请先阅读免责声明！）

**23：生成包关联图表**

难度：简单

应用程序：Debian

Debian系统最受争议的是它能安装一个包，并自动满足其附属（dependency）。如果你想要以图形形式表现这些包之间的关系（这对于了解系统如何协调工作是非常有用的），你可以使用Debian中非自由的Graphviz包（apt-get install graphviz）然后使用如下命令：

1. apt-cache dotty **>** debian.dot

这个命令会生成图片文件并存入dotty：

1. dotty debian.dot

**24：卸载忙碌状态的驱动**

难度：简单

应用程序：bash

你应该很熟悉这种情况-尝试去卸载一个驱动，但是系统却告诉你它处于忙碌状态。但是什么程序在占用呢？一个短小的单行代码将告诉你：

1. lsof +D /mnt/windows

这将返回命令和进程ID以及任何正在访问/mnt/windows目录的任务。接下来你就可以定位它们，或者使用kill命令来终结它们。

**25：文本文件的转换**

难度：简单

应用程序：recode

recode是一个小而实用并且可以帮你在不同平台上使用文本文件时节省大量精力的工具。它的缺陷主要是源于换行符。一些系统使用一个换行字符，而其他的一些系统使用回车。更多其他的系统两个都用。最后的结果就是，如果你从一个平台移动文本文件到另一个平台上，你会有太多或太少的换行符，还会有很多奇怪的字符残留。

然而，recode命令的参数是有一些神秘的，所以为什么不把这种技巧和HACK 26结合到这个功能中，然后建立一些有用的别名：

1. alias dos2unix='recode dos/CR-LF..l1'
2. alias unix2win='recode l1..windows-1250'
3. alias unix2dos='recode l1..dos/CR-LF'

|  |
| --- |
| ​ |

还有很多recode选项-它们确实可以转换大量字符间的设置。查阅man手册来获取更多信息。

**26：只列出今天的文件**

难度：简单

应用程序：Various

你可能熟悉这个问题，今天早些时候你创建了一个文本文档，而现在立刻就需要它。然而，你想不起来你给它起了个什么可笑的名字，作为一个熟练的geek，你的家文件夹被836个不同的文件填满了。那你要如何找到它呢？哈哈，这里有很多方法，但是这个小技巧告诉你管道和结合使用两个强力的shell命令是多么有效：

1. ls -al --time-style=+%D | grep `date +%D`

ls命令的参数使得这个日期戳以一种特殊的方式呈现。这个狡猾的位就是这样传递给grep来输出的。grep的参数是它自己的命令（因为是倒引号所以执行），用来将当前的日期替代成匹配的字符。你可以轻易的修改它来搜索其他特别的日期、时间、文件大小或者任何东西。结合它与HACK 26是非常节省输入的。

**27：避免常见的拼写错误和长命令**

难度：简单

应用程序：Shell

这个alias命令对于设立长命令的快捷键是有用的，或者说是聪明的事情。HACK 25中，我们可以通过以下命令来得到一个新命令，lsnew：

1. alias lsnew=" ls -al --time-style=+%D | grep `date +%D` "

然而alias还有其他的用处。例如，解决常见的拼写错误。有多少次你在改变父目录时忽略了空格？不用再苦恼了！

1. alias cd..="cd .."

作为一种选择，试试重写一些已经存在的命令怎么样？

1. alias ls="ls -al"

像我们一样节省一些按键，或者你总是想要完成清单。

要为每一个会话激活这些快捷键，只用添加alias命令到你的用户里家文件目录中的.bashrc文件中。

**28：更改Mozilla的秘密设置**

难度：简单

应用程序：Mozilla

如果你想要改变Mozilla操作，但是在首选项中并没有提供可以点击的选项来帮助你，这里有一个特殊的模式可以让你在Mozilla中激活，这样你就能改变一切了。要访问它，在地址栏中输入这些：

1. about:config

这样你就可以改变表中文件里每一个你感兴趣的设置的值。

其他有趣的模式包括总体信息（about：），插件详情（about:plugins），开发者信息（about:credits）和一些总体情况（about:mozilla）。

**29：一个星星的背景**

难度：简单

应用程序：KStars

你可能已经玩过KStars了，但是如何创造一个每次当你启动时就更新的KStars背景图呢？

KStars可以和–dump switch一起运行，它可以从你的启动设置中打印出一幅图片，但一点也不会增加GUI的负载。你可以写一个脚本来运行它生成一个每天都更换的桌面图片（或者你可以只是使用这个方法生成图片）。

像这样使用KStars：

1. kstars --dump --width 1024 --height 768 --filename = ~/kstarsback.png

你可以把它添加到~/.kde/Autostart文件夹的一个脚本中，一启动就开始运行。在浏览器中找到这个文件，把它拖到桌面上然后选择“设置为墙纸”，将它作为一个随机生成的背景图来使用。

**30：直接打开一个SVG**

难度：简单

应用程序：Inkscape

你可以通过shell运行Inkscape，然后直接通过一个URL来立刻编辑一张图片。只用输入：

1. inkscape http://www.somehost.com/graphic.svg

记得要另存为一下！

**31：不用编辑器来进行编辑**

难度：中等

1. cut -c 8-15 file1 **>** file2

应用程序：好多

通常非常长的文件是很难用文本编辑器来进行操作的。如果你不时地要做这件事，可能你会发现用某些趁手的命令行工具会更快一些，就像下面这个例子一样。

从文件file1中打印第一列和第三列到文件file2中，我们可以使用awk：

1. awk '{print $1, $3}' file1 **>** file2

只输出file1中第八列到第十五列中的字符，我们可以使用cut：

1. cut -c 8-15 file1 **>** file2

要使用单词word2来替换文件file1中的单词word1，我们可以使用sed命令：

1. sed "s/word1/word2/g" file1 **>** file2

通常这是一个比打开文本编辑器更快得到结果的方式。

**32：只备份选中的文件**

难度：中等

应用程序：tar

想要使用tar来备份一个目录中的某一文件？你可以使用-T标志。首先，用你想要备份的文件来创建一个文件：

1. cat **>>** /etc/backup.conf
2. # /etc/passwd
3. # /etc/shadow
4. # /etc/yp.conf
5. # /etc/sysctl.conf
6. EOF

|  |
| --- |
| ​ |

然后运行tar，并使用-T标志指向刚刚创建的文件。

1. tar -cjf bck-etc-`date +%Y-%m-%d`.tar.bz2 -T /etc/backup.conf

现在你已经备份好了。

**33：在文件中合并列**

难度：中等

应用程序：bash

在文件中拆分列是非常简单的，合并它们稍微复杂一点。下面是一个简单的完成这项工作的脚本：

1. #!/bin/sh
2. length=`wc -l $1 | awk '{print $1}'`
3. count=1
4. [ -f $3 ] && echo "Optionally removing $3" && rm -i $3
5. while [ "$count" -le "$length" ] ; do
6. a=`head -$count $1 | tail -1`
7. b=`head -$count $2 | tail -1`
8. echo "$a$b" **>>** $3
9. count=`expr $count + 1`
10. done

|  |
| --- |
| ​ |

给这个脚本命名为merge.sh并且如下运行它：

1. chmod u+x merge.sh

现在，如果你想要合并file1和file2中的列到file3，下面是执行方法：

1. /path/to/merge.sh file1 file2 file3

/path/to必须被你文件系统中的merge.sh位置所替代。

**34：大小写敏感**

难度：中等

应用程序：bash

有些时候，一个词在不同操作系统间没有任何区别，在Linux中，“Command”和“command”是不同的东西。当从Windows移动文件到Linux中时会造成一些麻烦。tr是一个小的shell功能，可以用于改变一堆文件。

1. #!/bin/sh
2. for i in `ls -1`; do
3. file1=`echo $i | tr [A-Z] [a-z] `
4. mv $i $file1 2**>**/dev/null
5. done

|  |
| --- |
| ​ |

通过执行这个，FILE1和fiLe2将被分别重命名为file1和file2.

**35：Emacs中的宏指令**

难度：中等

应用程序：Emacs

当编辑文件时，你通常会发现那些任务都是冗长和重复的，所以为了节约你的时间，你应该录制一个宏指令。在Emacs中，你将必须经历以下步骤：

1.按下 Ctrl+X 来开始一个记录。

2.插入所有你想要的按键和命令。

3.当你结束时按下Ctrl+X。

现在，你可以用以下命令来执行：

1. Ctrl -u **<number>** Ctrl -x e

<number>是你想执行宏指令的次数。如果你输入的值为0，这个宏指令将被一直执行直到文件的结束。Ctrl -x e等同于Ctrl -u 1 Ctrl-x e。

**36：简单的垃圾邮件清理**

难度：中等

应用程序：KMail

垃圾邮件，或者不请自来的大批邮件，已经是个很普遍的问题，出于必要，现在几乎每个人都有某种形式的垃圾邮件保护。很多ISP包含垃圾邮件过滤，但是它并没有被设置的太过主动，而且通常很多都是简单的标记垃圾邮件，然后让它进来（ISP可不想因为丢失了你的邮件而被责备）。

所以，你在客户端上可能已经设置了反垃圾邮件的东西，要让它运行得更好，你可以写一些过滤器来移除被标记的垃圾邮件。标签是作为标题包含进来的。在KMail中，你可以只创建一个快速过滤器来将你的邮件装箱，或者干脆把它放到垃圾文件夹中。精确标题的使用将决定于你的ISP使用的软件，但是通常都是像SpamAssassin这样系统的类似X-Spam-Flag = YES。

在KMail中简单创建一个过滤器，选择Match Any of the Following（匹配下面的任何一个）然后输入标题细节和你需要的动作。应用过滤器到即将到来的邮件上，然后你的垃圾邮件的一半将不会再来打扰你了。

**37：阅读OOo文档但不使用OOo**

难度：中等

应用程序：OpenOffice.org（译注：一套跨平台办公室软件套件）

你曾经有没有过拿到一个OOo文档，但是没有OpenOffice.org来阅读？所以你就把它保存成普通的文本文件（.txt），但是被 StarOffice .sxw 格式所替代？这个文本文件可以得到拯救。首先，这个sxw文件是一个压缩文档，所以解压它：

1. unzip myfile.sxw

你想要的是名为’content.xml’的文件。不幸的是，它充满了xml的标签以至于非常难以辨认，所以要用一些Perl魔法将它们过滤出去：

1. cat content.xml | perl -p -e"s/**<**[^**>**]\***>**/ /g;s/\n/ /g;s/ +/ /;"

这可能丢失了很多格式，但是至少现在它是可以阅读的。

**38：搜索和执行**

难度：中等

应用程序：find

find命令并不只是在寻找文件方面很有用，它在处理以下问题时也是非常有用的。这里有一个简短的例子。

假设我们有很多的压缩文件，现在我们想找到所有的压缩文件：

1. find . -name '\*.gz'

将会定位出所有当前路径下的gzip文档。但是假设我们想要检查它们是否是有效文档，gunzip -vt选项将会帮我们实现这个，我们还可以机智地使用xargs结合这两个操作：

1. find . -name '\*.gz' | xargs gunzip -vt

**39：正确使用whois服务器**

难度：中等

应用程序：whois

whois命令在追捕Internet上的恶棍以及给他们提供服务的ISP都是非常有用的。不幸的是，有很多whois服务器，如果你的反搜索一个域名，通常你不得不用一个指定的他们使用的TLD。然而，一些whois代理将自动转送你的查询到正确的服务器上。其中之一可以在http://whois.geektools.com/上获得。

1. whois -h whois.geektools.complop.info

**40：驱动挂载到哪里了？**

难度：中等

应用程序：bash

人们常碰到的一个问题是，你刚接通电源时，就有太多的可挂载外设（USB驱动，flash内存卡，USB键驱动）消耗光了驱动？

实际上，所有的驱动都调用一个驱动——就像usb-storage，会在日志中存储很多有用的信息。尝试：

1. dmesg | grep SCSI

这将从dmesg输出过滤出指定的驱动规格。你可能发现一些这样的文本：

1. SCSI device sda: 125952 512-byte hdwr sectors (64 MB)

这个是说你的外设是在sda上。

**41：自动执行USB装置**

难度：高级

应用程序：hotplug脚本

当添加一个特殊的外设时想要运行一个指定的应用程序？USB hotplug后台进程可以帮你！当有USB装置添加到系统中时，这个服务就会得到通知。对于需要内核驱动的装置，hotplug后台进程将在/etc/hotplug/usb/下调用一个同名的脚本，例如，这里的一个名为usb-storage的脚本。你可以直接在这个脚本的后面添加你自己的命令（或者更好的是，在它的最后添加一行来让脚本在别的地方执行）。那么你就可以播放音乐，autosync文件（译注：影像文件），搜寻图片或者其他什么的。

对于不依赖于内核驱动的装置，有一个检查表，可以用来匹配USB产品和制造商ID。很多发行版已经设置好去做一些事情，但是你可以依据你的需要定制这些脚本。到http://jphoto.sourceforge.net/ selected=sync去看一个我们能做什么的例子。

**42：重命名和调整图片大小**

难度：高级

应用程序：bash

很喜欢你的新相机但是不能容忍那个糟糕的名字？你也想要准备发布它们到网上吗？没问题，一个简单的bash脚本真是你所需要的：

1. #!/bin/sh
2. counter=1
3. root=mypict
4. resolution=400x300
5. for i in `ls -1 $1/\*.jpg`; do
6. echo "Now working on $i"
7. convert -resize $resolution $i ${root}\_${counter}.jpg
8. counter=`expr $counter + 1`
9. done

|  |
| --- |
| ​ |

保存这个脚本到一个名为picturename.sh的文件中，然后用一下命令执行它：

1. chmod u+x picturename.sh

然后在你的某个路径下存储它。现在，如果你在directory /path/to/pictdir下有一堆.jpg文件，所有你需要做的就是执行这个：

1. picturename.sh /path/to/pictdir

而在当前的目录下你会发现 mypict\_1.jpg, mypict\_2.jpg之类的，它们是你初始图片调整大小后的版本。你可以根据你的需要改变脚本，或者，你只是期待一个超级简单的图片大小调整，试着看看mogrify命令和它的-geometry参数。

**43：安全登出**

难度：简单

应用程序：bash

当你在一台共享机器上使用一个控制台，或者甚至只是在你自己的桌面使用控制台，都可能会发现当你登出时，屏幕仍然会显示谁已经登入正在做什么的痕迹。大多数发行版本将会清屏，但是有些不会。你可以通过编辑你的 ~/.bash\_logout文件然后增加这个命令来解决问题：

1. clear

你也可以添加任何其他的有用命令在这里。

**44：传输文件但不使用ftp或者scp**

难度：简单

应用程序：netcat

需要传输一个目录给其他服务器但是没有FTP或者SCP使用权？那么这个小技巧将会帮助你使用netcat功能。在目的服务器上运行：

1. nc -l -p 1234 | uncompress -c | tar xvfp -

并且在发送服务器上运行：

1. tar cfp - /some/dir | compress -c | nc -w 3 [destination] 1234

现在你就可以不需要FTP并且不需要root权限，就能传送文件了。

**45：备份一个Debian包清单**

难度：简单

应用程序：Debian

如果你正在运行Debian并且找不到你正在运行的包，为你当前正在安装的包做一个备份是非常有用的。你可以运行如下得到一个清单：

1. dpkg --get-selections **>** debianlist.txt

这将把全部的清单放到debianlist.txt中，这样你就可以通过如下代码在不同的计算机上安装相同的包。

1. dpkg --set-selections **<debianlist.txt**

当你复制你的系统到一个新的计算机上时，你要记住你还需要从/etc中复制配置文件。

要实际安装这些选项，使用如下指令：

1. apt-get -u dselect-upgrade

**46：强化ssh**

难度：简单

应用程序：ssh

尽管SSH是一个非常安全的连接服务器的方式，这里还有两个简单的改变，可以让你进一步增强它的安全性。首先，你当然不希望别人直接以root用户登入——他们应该以普通用户登陆，然后使用su命令来转换。你可以在/etc/ssh/ssh\_config 文件中通过添加如下代码改变这个件事：

1. PermitRootLoginno

现在唯一能得到root特权的方式是通过su，这意味着入侵者现在需要破解两个密码来得到全部的使用权。当你编辑这个文件时，找到这样一行：

1. Protocol 2, 1

然后把它改为如下：

1. Protocol 2

这将消除最初的SSH协议中的撤销选项，现在被认为是非常易于攻击的。

**47：停止回应ping**

难度：简单

应用程序：sysctl

ping在找出网络拓扑结构时是非常有用的命令，缺点是它只能做这件事，而且会使得网络上的黑客们更容易瞄准现场服务器。但是你可以告诉Linux忽略所有的ping服务器将不会回应。这里有很多的方法来实现它，但是最好的是使用sysctl。关掉ping回应：

1. sysctl -w net.ipv4.icmp\_echo\_ignore\_all=1

想要打开它，使用：

1. sysctl -w net.ipv4.icmp\_echo\_ignore\_all=0

关掉ping对你来说是非常严重的，看看接下来的技巧。

**48：降低ping速率**

难度：简单

应用程序：sysctl

可能你想要保持服务器回复ping的功能，但是又想从被熟知的“ping flood”（封包洪流）的攻击中保护自己。那么你要如何完成这一壮举呢？最简单的方法是降低回应ping服务器的速率。它们仍然是有效的，但不会使服务器过载。

1. sysctl -w net.ipv4.icmp\_echoreply\_rate=10

当回复被当作一个信号地址发送时，这将会降低速率。

**49：当登出的时候清理KDE**

难度：简单

应用程序：bash

在Windows中有很多程序会做一些清除网络缓存，移除临时文件事情。Linux上如果也可以这样是不是很cool呢？使用KDE，你甚至不需要安装任何新的软件，startkde脚本将会自动的执行你放在特殊地方的脚本。

首先，你需要在你的.kde目录下创建一个命令shutdown的目录：

1. mkdir /home/username/.kde/shutdown

现在创建一个脚本，用来在关机时执行一些你想要的事情。这里是一个例子：

1. #!/bin/bash
2. #clear up temp folder
3. rm -rf ~/tmp/\*
4. #clear out caches
5. rm -rf ~/.ee/minis/\*
6. rm -rf ~/.kde/share/cache/http/\*
7. # delete konqueror form completions
8. rm ~/.kde/share/apps/khtml/formcompletions

现在确保你设置了正确的权限：

1. chmod ug+x ~/.kde/shutdown/cleanup.sh

（或者随便你叫它什么）。可以清除敏感文件，也可以通过将脚本放在你的默认KDE文件夹中有一个名为shutdown的子文件夹里，给所有用户一个全局的关机脚本。找出你的默认KDE目录在哪里，试一下：

1. kde-config --path exe

**50：无密码的ssh**

难度：中等

应用程序：ssh

厌烦了每次登陆你的服务器时都要输入密码？ssh也支持密钥，所以当你登陆桌面时你只能输入你的密码。在你的桌面机器上生成一个密钥对：

1. ssh-keygen -t dsa -C your.email@ddress

为你的密钥输入一个口令。这会将密钥放到 ~/.ssh/id\_dsa and the public key in ~/.ssh/id\_dsa.pub中。现在看看你是否有了一个正在运行的ssh-agent。

1. echo $SSH\_AGENT\_PID

如果安装了ssh-agent，大部分窗口管理器都会自动运行它。如果没有安装，开启一个：

1. eval $(ssh-agent)

现在，告诉这个代理你的密钥：

1. ssh-add

然后输入你的口令。你需要在每次登陆时都输入口令，如果你是使用X，尝试添加：

1. SSH\_ASKPASS=ssh-askpass ssh-add

到你的.xsession文件。（你可能需要安装ssh-askpass）现在为每个你需要登入的服务器都创建这个目录 ~/.ssh，然后复制这个文件~/.ssh/id\_dsa.pub into it as ~/.ssh/authorized\_keys 。如果你是手动开启ssh-agent，当你登出时，使用如下代码终止它：

1. ssh-agent -k

**51：通过ssh使用rsync**

难度：中等

应用程序：Shell

rsync使大型目录结构保持同步。通过SSH使用tar远程复制一个文件系统的一部分是非常理想的，sync甚至更适合保持两台机器之间的文件系统的同步。为了使用SSH运行一个rsyns，传递给它一个-e转换：

1. rsync -ave ssh greendome:/home/ftp/pub/ /home/ftp/pub/

注意从源端（在greendome上）来的文件说明后面的 / ，在源说明中，一个尾部的 / 告诉rsync这个目录的内容，但不是目录本身。为了将目录放入正在复制的内容的最高等级丢掉这个 /：

1. rsync -ave ssh bcnu:/home/six .

这将在 ~/six/目录下保持一个与bcnu:/home/six/上的任何东西同步的复制品。rsync默认只拷贝文件和目录，但是当源端移除了文件时并不会在目的端移除它们的拷贝。为了保持复制的精确性，包含–delete这个标志：

1. rsync -ave ssh--delete greendome:~one/reports .

现在当老的报表从greendome上的~one/reports/中移除了，在同步版本中，每次这个命令运行时，它们也将会从~six/public\_html/reports/中移除。如果你在corn中像这样运行一条命令，记得丢掉v。这将会使输出稳定（除非rsync在运行中出现问题，在这种情况下你将会收到一封有错误输出的邮件）。使用SSH作为你的rsync运输方式，这样做的好处是使数据在网络传输过程中加密，并可以方便使用任何你已经用SSH客户端密钥建立的信任关系。

**52：资源扫描**

难度：中等

应用程序：nmap

例如，简单的TCP连接扫描可以在不需要root特权时进行：

一般来说，当人们想到nmap时，都假设nmap是用来进行一些违法的网络侦查，并为恶意攻击做准备的。但是作为一个强大的工具，nmap可以成为一个白帽子，做一些比入侵网络有意义得多的事。

1. nmap rigel

nmap也可以通过指定范围或者CIDR符号来扫描IP地址：

1. nmap 192.168.0.1-254
2. nmap 192.168.0.0/24

nmap如果以root身份运行，可以提供更多的信息。当它以root身份运行时，通过使用-O标志，它可以使用特殊的数据包来决定远程机器的操作系统。此外，你可以通过使用-sS标志来实现半开的TCP扫描。nmap将会发送一个SYN数据包到远程的主机，然后等待接收它回应的ACK。如果nmap收到一个ACK，它就知道这个端口是开放的。

这不同于普通的TCP三次握手，客户端会发送一个SYN包，如果它收到了初始的服务器ACK，客户端随后会将一个ACK送回服务器。攻击者们时常使用这个选项来避免使他们的扫描登陆到远程机器上。

1. nmap -sS -O rigel
2. Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
3. Interesting ports on rigel.nnc (192.168.0.61):
4. (The 1578 ports scanned but not shown below are in state: filtered)
5. Port State Service
6. 7/tcpopen echo
7. 9/tcpopen discard
8. 13/tcp open daytime
9. 19/tcp open chargen
10. 21/tcp open ftp
11. 22/tcp open ssh
12. 23/tcp open telnet
13. 25/tcp open smtp
14. 37/tcp open time
15. 79/tcp open finger
16. 111/tcpopen sunrpc
17. 512/tcpopen exec
18. 513/tcpopen login
19. 514/tcpopen shell
20. 587/tcpopen submission
21. 7100/tcp open font-service
22. 32771/tcpopen sometimes-rpc5
23. 32772/tcpopen sometimes-rpc7
24. 32773/tcpopen sometimes-rpc9
25. 32774/tcpopen sometimes-rpc11
26. 32777/tcpopen sometimes-rpc17
27. Remote operating system guess: Solaris 9 Beta through Release on SPARC
28. Uptime 44.051 days (since Sat Nov1 16:41:50 2003)
29. Nmap run completed -- 1 IP address (1 host up) scanned in 166 seconds

|  |
| --- |
| ​ |

伴随着OS探测的激活，nmap可以确认OS是Solaris，但是现在你还可以知道，它可能是运行在SPARC处理器上的Version 9。

一个强大的功能是可以被用于和你的nmap的XML输出性能保持联系。使用-oX命令行转换而激活，就像这样：

1. nmap -sS -O -oX scandata.xml rigel

这在扫描IP地址范围或者你全部的网络时是特别有用的，你可以将所有通过扫描收集起来的信息放到一个XML文件里，并经过解析后插入到数据库中。这里是一个开放的端口的XML入口：

1. **<port**protocol="tcp"portid="22"**>**
2. **<state**state="open"**/>**
3. **<service**name="ssh"method="table"conf="3"**/>**
4. **</port>**

|  |
| --- |
| ​ |

nmap是一个强大的工具。通过使用它的XML输出性能，一点点的脚本，还有一个数据库，你可以创造一个更强大的可以在你的网络上监测未授权的服务和机器的工具。

**53：备份你的bootsector(引导扇区)**

难度：高级

应用程序：Shell

妨碍引导装入，双启动和很多其他下热闹的程序会留给你一个混乱的引导扇区。为什么不在你可以备份的时候给它创建一个备份呢：

引导装载器混乱、双启动及许多其他的可怕的进程可能造成乱七八糟的启动区。

1. dd if=/dev/hda of=bootsector.img bs=512count=1

很显然，你应该改变这个装置来显示你的boot驱动（有可能是SCSI的sda）。还要非常小心，别把事情搞砸——你可能会轻而易举地毁掉你的驱动！恢复时使用：

1. dd if=bootsector.img of=/dev/hda

**54：保护日志文件**

难度：高级

应用程序：很多

在一个入侵过程中，一个入侵者将会很可能在很多系统日志中留下能反映他行为的迹象：一个有价值的审计追踪应该被保护。没有可靠的日志，就很难发现攻击者是如何侵入，或者是从哪里来的攻击。这个信息在分析事件、联系相关方面并给予反馈上，是很关键的。但是，如果入侵成功，什么才能阻止他来移除他的这些错误行为的痕迹呢？

这就是文件属性能发挥作用、挽回败局的地方（或者至少使事情没那么糟）。Linux和BSD都有给文件和目录分配额外的属性这个功能。这不同于标准的Unix权限方案中，系统为所有的用户提供的普遍的属性设置，而且它们比文件权限或者ACL都要更深层次影响文件访问。

在Linux中，你可以看到并且通过lsattr和chattr命令分别修改一个给出文件的设置。在写这篇文章时，Linux的文件属性只能在你使用ext2和ext3时获得。这还有些XFS和ReiserFS的关于获得属性支持的内核补丁。一个保护日志文件的有用的属性是append-only。当这个属性被设置后，文件不能被删除，而且写操作只能被允许在文件的最后追加。

在Linux下要设置append-only标志，运行这条命令：

1. chattr +afilename

看看+a属性是如何工作的：创建一个文件然后设置它的append-only属性：

1. touch /var/log/logfile
2. echo "append-only not set" **>** /var/log/logfile
3. chattr +a /var/log/logfile
4. echo "append-only set" **>** /var/log/logfile
5. bash: /var/log/logfile: Operation not permitted

企图第二次写文件失败，因为它将覆盖这个文件。然而，在文件的末尾追加仍然是被允许的：

1. echo "appending to file" **>>** /var/log/logfile
2. cat /var/log/logfile
3. append-only not set
4. appending to file

显而易见，一个获得root特权的侵入者能意识到使用了文件属性，并且可以通过运行chattr -a来移除append-only标志。为了防止这种情况，我们需要禁用移除append-only属性的性能，在Linux下使用功能机制完成这个。

Linux功能模型分割特权给全能的root账户，并且允许你有选择的禁用它们。为了防止一个用户移除一个文件的append-only属性，我们需要移除CAP\_LINUX\_IMMUTABLE功能。当体现在当前运行系统中时，这个功能允许append-only属性被修改。为了当前系统中获得修改这个功能的设置，我们将简单有效地调用lcap(http://packetstormsecurity.org/linux/admin/lcap-0.0.3.tar.bz2).

为了解压和编译这个工具，运行这条命令：

1. tar xvfj lcap-0.0.3.tar.bz2 && cd lcap-0.0.3 && make

然后，不允许修改append-only标志，运行：

1. ./lcap CAP\_LINUX\_IMMUTABLE
2. ./lcap CAP\_SYS\_RAWIO

|  |
| --- |
| ​ |

第一条命令移除了改变append-only标志的功能，然后第二条移除了原始I/O的功能。这是很有必要的，因为这样就能保护文件不被访问块设备时驻留在其上的东西修改。这阻止了访问/dev/mem和/dev/kmem，有可能给入侵者提供恢复CAP\_LINUX\_IMMUTABLE功能的漏洞。为了在boot上移除这项功能，添加之前的两条命令到你的系统启动脚本中（例如/etc/rc.local）。你应该确保这个功能在boot命令中被移除了，防止其他启动脚本出问题。一旦lcap移除了内核功能，就只能通过重启系统来恢复了。

在做这些之前，你应该明白，给你的日志文件添加append-only标志，将会造成日志循环脚本失败。然而，做这件事将会极大的提升你的审计追踪的安全性。安全性的提升在突发事件中会被证明是很有用的。

**55：自动加密连接**

难度：高级

应用程序：FreeS/WAN

一个FreeS/WAN支持的特别cool的功能是，当其他主机运行FreeS/WAN时随机加密。这允许FreeS/WAN在所有支持随机加密的主机间透明传输。为了这个，每台主机必须使用FreeS/WAN生成一个公钥，这个公钥可以存储于那台主机的一个DNS TXT记录中。当一台主机要与另一个主机建立一个随机加密，希望初始化一个加密连接时，将会通过DNS查询这台主机的公钥并且初始化这个连接。

一开始，你想使用这个功能时，需要为每一台主机生成一个公钥。你可以通过运行如下命令来实现它：

1. ipsec newhostkey --output /tmp/`hostname`.key

现在你需要添加通过那条命令创建的文件的内容到/etc/ipsec.secrets中。

1. cat /tmp/`hostname`.key **>>** /etc/ipsec.secrets

然后，你需要生成一个TXT记录来存放你的DNS区域。你可以通过运行如下命令来实现它：

1. ipsec showhostkey --txt @colossus.nnc

现在添加这条记录到你的空间然后重载它。你会证实DNS已经通过这条命令正确的运行了：

1. ipsec verify
2. Checking your system to see if IPsec got installed and started correctly
3. Version check and ipsec on-path[OK]
4. Checking for KLIPS support in kernel [OK]
5. Checking for RSA private key (/etc/ipsec.secrets) [OK]
6. Checking that pluto is running [OK]
7. DNS checks.
8. Looking for TXT in forward map: colossus [OK]
9. Does the machine have at least one non-private address [OK]

|  |
| --- |
| ​ |

现在只需要重启FreeS/WAN-你现在应该能连接到任何支持随机加密的主机上了。但是如果其他的主机想要连接你该怎么办？要授权连接，你需要在你的DNS反向查找区域为你的机器创建一个TXT记录。

你可以通过运行一个简单的命令来生成一条记录：

1. ipsec showhostkey --txt 192.168.0.64

为你的子网添加这条记录到反向查找区域，然后其他的机器就可以和你的机器初始化随机加密了。伴随着随机加密的使用，所有主机间的传输将被自动加密，同时保护所有的服务。

**56：消除二进制suid**

难度：中等

应用程序：find

如果你的服务器有比你自身多的shell用户，你应该在你的系统上定期的审核setuid和setgid的二进制文件。可能你会惊讶你找到了那么多文件。这里有一个搜寻所有setuid或者setgid的文件的命令位设置：

1. find / -perm +6000 -type f -exec ls -ld {} \; **>** setuid.txt &

这将会创建一个名为setuid.txt的文件，包含了所有当前系统上匹配的文件的细节。要移除任何你不用的工具的s位，输入：

1. chmod a-s program

**57：Mac过滤Host AP**

难度：高级

应用程序：iwpriv

你可以在链路层使用iptables或者ebtables执行MAC过滤，那比让Host AP为你做这些更安全。这不仅仅是阻碍指向你的网络传输，也是预防邪恶的人关联你的站。这能帮助组织某些人给你的其他无线客户端造成麻烦的可能，即使他们没有更多的网络通道。

当使用MAC过滤时，很多人会列出一个他们希望允许的无线装置清单，然后拒绝所有其他的。使用iwpriv命令完成这个功能：

1. iwpriv wlan0 addmac 00:30:65:23:17:05
2. iwpriv wlan0 addmac 00:40:96:aa:99:fd
3. ...
4. iwpriv wlan0 maccmd 1
5. iwpriv wlan0 maccmd 4

|  |
| --- |
| ​ |

addmac指示添加一个MAC地址到内部表。只要你喜欢你就可以通过执行更多的addmac命令添加更多的MAC地址到表中。然后你需要告诉Host AP对你已经建好的表做什么。maccmd 1命令告诉Host AP就像使用一个“允许”清单一样使用这个表，并且拒绝所有其他的MAC地址来关联。最终，maccmd 4命令去除掉所有的关联客户端，强制使它们重新关联。对表中的客户端来说是自动发生的，但是其他的任何企图关联的客户端都是被拒绝的。

有时，你只需要禁止一个或者两个捣蛋鬼，而不是设置一个详细的被允许的硬件规则。如果你需要禁止两三个MAC地址而允许其他所有的地址，尝试这个：

1. iwpriv wlan0 addmac 00:30:65:fa:ca:de
2. iwpriv wlan0 maccmd 2
3. iwpriv wlan0 kickmac 00:30:65:fa:ca:de

像以前一样，你可以不限次数的使用addmac，只要你乐意。Maccmd2命令会将规则设置为“拒绝”，而如果该MAC恰巧已经关联，kickmac会立刻将特定的MACboot。这也许比只为了禁止一个捣蛋鬼而消除所有关联然后让他们重新邻接要好一些。顺便说一下，如果你想要消除所有的MAC过滤，尝试maccmd 0。

如果你错误的输入了一个MAC地址，你可以像使用addmac一样使用delmac命令，然后它（可以预见）就会从表中删掉这个给出的MAC地址。你有时会需要完全清除当前的MAC表但是保留它当前的规则，使用这条命令：

1. iwpriv wlan0 maccmd 3

最后，你可以通过使用/proc浏览正在运行的MAC表：

1. cat /proc/net/hostap/wlan0/ap\_control

iwpriv程序操纵正在运行的Host AP驱动，但是重启后不保留设置。一旦你对你的MAC过滤表感到满意，请确保你在一个rc脚本中添加了相关命令在启动时间运行。

注意，未关联的客户端仍然可以监听网络传输，所以MAC过滤实际上只能预防一点点窃取。为了与被动的监听技巧搏斗，你需要加密你的数据。