CTRL + W：删除光标位置前的单词

CTRL + U：清空行

↑，↓方向键：查看命令历史

Tab：自动补全文件名、目录名和命令等等

CTRL + R：搜索先前使用的命令

CTRL + C：中止当前命令

CTRL + D：退出登录 Shell

ESC + T：调换光标前的两个单词

系统信息

arch 显示机器的处理器架构(1)

name -m 显示机器的处理器架构(2)

name -r 显示正在使用的内核版本

dmidecode -q 显示硬件系统部件-(SMBIOS/DMI)

hdparm -i /dev/hda 罗列一个磁盘的架构特性

hdparm -tT /dev/sda 在磁盘上执行测试读取操作

cat /proc/cpuinfo 显示 CPU info 的信息

cat /proc/interrupts 显示中断

cat /proc/meminfo 校验内存使用

cat /proc/swaps 显示哪些 swap 被使用

cat /proc/version 显示内核的版本

cat /proc/net/dev 显示网络适配器及统计

cat /proc/mounts 显示已加载的文件系统

lspci -tv 罗列 PCI 设备

lsusb -tv 显示 USB 设备

date 显示系统日期

cal 2007 显示 2007 年的日历表

date 04127002007.00 设置日期和时间-月日时分年.秒

clock -w 将时间修改保存到 BIOS

关机（系统的关机、重启以及登出）

shutdown -h now 关闭系统（1）

init 0 关闭系统（2）

telinit 0 关闭系统（3）

shutdown -h hours:minutes & 按预订时间关闭系统

shutdown -c 取消按预订时间关闭系统

shutdown -r now 重启（1）

reoot 重启（2）

logout 注销

文件和目录

cd /home 进入'/home'目录

cd .. 返回上一级目录

cd ../.. 返回上两级目录

cd 进入个人的主目录

cd ~user1 进入个人的主目录

cd - 返回上次所在的目录

pwd 显示工作路径

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

tree 显示文件和目录由根目录开始的树形结构（1）

lstree 显卡文件和目录由根目录开始的树形结构（2）

mkdir dir1 创建一个叫做'dir1'的目录'

mkdir dir1 dir2 同时创建两个目录

mkdir dir1/dir2 创建一个目录树

rm -f file1 删除一个叫'file1'的文件

rmdir dir1 删除一个叫‘dir1’的目录

rm -rf dir1 删除一个叫‘dir1’的目录并同时删除其内容

rm -rf dir1 dir2 同时删除两个目录及它们的内容

mv dir1 new_dir 重命名/移动一个目录

cp file1 file2 复制一个文件

cp dir/*. 复制一个目录下的所有文件到当前工作目录

cp -a /tmp/dir1. 复制一个目录到当前工作目录

cp -a /tmp/dir2 复制一个目录

ln -s file1 lnk1 创建一个指向文件或目录的软链接

ln file1 lnk1 创建一个指向文件或目录的物理链接

touch -t 0712250000 file1 修改一个文件或目录的时间戳-(YYMMDDhhmm)

file file1 outputs the mime type of the file as text

iconv -l 列出已知的编码

iconv -f fromEncoding -t toEncoding inputFile>outputFile creates a new from the given input file by assuming it is encoded in fromEncoding and converting it to toEncoding. find. -maxdepth 1 -name *.jpg -print -exec convert"{}" -resize 80x60"thumbs/{}"\;batchresize files in the current directory and send them to a thumbnails directory (requires convert from I magemagick)

文件搜索

find /-name file1 从 '/'

开始进入根文件系统搜索文件和目录

find /-user user1 搜索属于用户 'user1' 的文件和目录

find /home/user1 -name *.bin 在目录 '/homi/user1' 中搜索带有 'bin' 结尾的文件

find /usr/bin -type f -atime +100 搜索在过去 100 天内未被使用过的执行文件

find /usr/bin -type f -mtime -10 搜索在 10 天内被创建或者修改过的文件

find /-name *.rpm -exec chmod 755'{}'\; 搜素以 '.rpm' 结尾的文件并定义其权限

find /-xdev -name *.rpm 搜索以 '.rpm' 结尾的文件，忽略光驱、键盘等可移动设备

locate *.ps 寻找以 '.ps' 结尾的文件 -先运行 'updatedb' 命令

whereis halt 显示一个二进制文件、源码或 man 的位置

挂载一个文件系统

mount /dev/hda2/mnt/hda2 挂载一个叫 hda2 的盘 -确定目录 '/mnt/hda2' 已经存在

umount /dev/hda2 卸载一个叫做 hda2 的盘 -先从挂载点 ‘/mnt/hda2’ 推出

fuser -km /mnt/da2 当前设比繁忙时强制卸载

umount -n /mnt/hda2 运行卸载操作而不写入 /etc/mtab 文件-当文件为只读或当磁盘写满时非常有用

mount /dev/fd0 /mnt/floppy 挂载一个软盘

mount /dev/cdrom /mnt/cdrom 挂载一个 cdrom 或 dvdrom

mount /dev /hdc /mnt/cdrecorder 挂载一个 cdrw 或 dvdrom

mount -o loop file.iso /mnt/cdrom 挂载一个文件或 ISO 镜像文件

mount -t vfat /dev/hda5 /mnt/hda5 挂载一个 windows FAT32 文件系统

mount /dev/sda1 /mnt/usdisk 挂载一个 usb 捷盘或闪存设备

mount -t smbfs -o username=user,password=pass //WinClient/share /mnt/share

挂载一个 windows 网路共享

磁盘空间

df -h 显示已挂载的分区列表

ls -lSr |more 以尺寸大小排列文件和目录

du -sh dir1 估算目录 ‘dir1’ 已经使用的磁盘空间

du -sk * | sort -rm 以容量大小为依据次显示文件和目录的大小

rpm -q -a --qf '%10{SIZE}n' | sort -kl,1n 以大小为依据次显示已安装的 rpm 包所使用的空间(fedora, redhat 类系统)

dpkg-query -W -f='${Installed-Size;10}t${Package}n' |sort -k1,1n 以大小为依据显示已安装的 deb 包所使用的空间（ ubuntu,debian 类系统 ）

用户和群组

groupadd group_name 创建一个新用户组

groupdel group_name 删除一个用户组

groupmod -n new_group_name old_group_name 重命名一个用户组

useradd -c "Name Surname" -g admin -d /home/user1 -s /bin/bash user1 创建一个属于“admin”用户组的用户

useradd user1 创建一个新用户

userdel -r user1 删除一个用户（ ‘-r’ 排除主目录）

usermod -c"User FTP" -g system -d /ftp/user1 -s/bin/nologin user1 修改用户属性

passwd 修改口令

passwd user1 修改一个用户的口令（之允许 root 执行）

chage -E 2005-12-31 user1 设置用户口令的失效期限

pwck 检查 ‘/etc/qasswd’ 的文件格式和语法修正以及存在的用户

grpck 检查 ‘/etc/pallwd’ 的文件格式和语法修正以及存在的群组

newgrp group_name 登录进一个新的群组以改变新

创建文件的预设群组

文件的权限-使用“+”设置权限，使用“-”用于取消

ls -lh 显示权限

ls /tmp | pr -T5 -W$COLUMNS 将终端划分成 5 栏显示

chmod ugo+rwx directory1 设置目录的所有人（u）、群组（g）以及其他人（o）以读（r）、写（w）和执行（x）的权限

chmod go-rwx directory1 删除群组（g）与其他人（o）对目录的读写执行权限

chmown user1 file1 改变一个文件的所有人属性

chown -R user1 directory1 改变一个目录的所有让人属性并同时改变目录下所有文件的属性

chgrp group1 file1 改变文件的群组

chown user1:group1 file1 改变一个文件的所有人和群组属性

find /-perm -u+s 罗列一个系统中所有使用了 SUID 控制的文件

chmod u+s /bin/file1 设置一个二进制文件的 SUID 位-与性该文件的用户也被赋予和所有者同样的权限

chmod u-s /bin/file1 禁用一个二进制文件的 SUID 位

chmod g+s /home/public 设置一个目录的 SGID 位-类似 SUID，不过这是针对目录的

chmod g-s /home/public 禁用一个目录的 SGID 位

chmod o+t /home/public 设置一个文件的 STIKY 位-只允许合法所有人删除文件

chmod o-t /home/public 禁用一个目录的 STIKY 位

文件的特殊属性-使用“+”设置权限，使用“-”用于取消

chattr +a file1 只允许追加方式写文件

chatr +c file1 允许这个文件能被内核自动压缩/解压

chattr +d file1 在运行文件系统备份时，dump 程序将忽略这个文件

chattr +i file1 设置成不可改变的文件，不能被删除、修改、重命名或者链接

chattr +s file1 允许一个文件被安全地删除

chattr +S file1 一旦应用程序对这个文件执行了写操作，使系统立刻把修改的结果写到磁盘

chattr +u file1 若文件被删除，系统会允许你在以后恢复这个被删除的文件

lattr 显示特殊的属性

打包和压缩文件

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 压缩以额叫做‘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‘file.rar’的包

rar a file.rar file1 file2 dir1 同时压缩‘file1’，‘file2’以及目录‘dir1’

rar x file1.rar 解压 rar 包

unuar x file1.rar 解压 rar 包

tar -cvf archive.tar file1 创建一个非压缩的 tarbal1

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

RPM 包-（Fedora,Redhat 及类似系统）

rpm -ivh package.rpm 安装一个 rpm 包

rpm -ivh --nodeeps package.rpm 安装一个 prm 包而忽略依赖关系警告

rpm -U package.rpm 更新一个 rpm 包但不改变其配置文件

rpm -F package.rpm 更新一个确定已安装的 rpm 包

rpm -e package_name.rpm 删除一个 rpm 包

rpm -qa 显示系统中所有已安装的 rpm 包

rpm -qa | grep httpd 显示所有名称中包含“httpd”字样的 rpm 包

rpm -qi package_name 获取一个已安装包的特殊信息

rpm -qg "System Environment/Daemons"显示一个组件的 rpm 包

rpm -ql package_name 显示已经安装的 rpm 包提供的文件列表

rpm -qc package_name 显示一个已经安装的 rpm 包提供的配置文件列表

rpm -q package_name --whatrequires 显示与一个 rpm 包存在依赖关系的列表

rpm -q package_name --whatprovides 显示一个 rpm 包所占的体积

rpm -q package_name --scripts 显示在按钻根/删除期间所执行的脚本 l

rpm -q package_name --changelog 显示一个 rpm 包的修改历史

rpm -qf /etc/httpd/conf/httpd.conf 确认所给的文件由哪个 rpm 包所提供

rpm -qp package.rpm -l 显示由一个尚未安装的 rpm 包提供的文件列表

rpm --import /media/cdrom/RPM-GPG-KEY 导入公钥数字证书

rmp --checksig package.rpm 确认一个 rpm 包的完整性

rpm -qa gpg-pubkey 确认已安装所有 rpm 包的完整性

rpm -V package_name 检查文件尺寸、许可、类型、所有者、群组、MD5 检查以及最后修改时间

rpm -Va 检查系统中所有已安装的 rpm 包-小心使用

rpm -Vp package.rpm 确认一个 rpm 包还未安装

rpm2cpio package.rpm | cpio --extract --make-directories *bin* 从一个 rpm 包运行可执行文件

rpm -ivh /usr/src/redhat/RMS/arch /package.rpm 从一个 rpm 源码安装一个构建好的包

rpmbuild --reuild package_name.src.rpm 从一个 rpm 源码构建一个 rpm 包

YUM 软件包升级器 -（Fedora,RedHat 及类似系统）

yum install package_name 下载并安装一个 rpm 包

yum localinstall package_name.rpm 将安装一个 rpm 包，使用你自己的软件仓库为你解决所有依赖关系

yum update package_name.rpm 更新当前系统中所有的 rpm 包

yum update package_name 更型一个 rpm 包

yum remove package_name 删除一个 rpm 包

yum list 列出当前系统中安装的所有包

yum search package_name 在 rpm 仓库中搜寻软件包

yum clean packages 清理 rpm 缓存删除下载的包

yum clean headers 删除所有头文件

yum clean all 删除所有缓存的包和头文件

DEB 包（Debian,Ubuntu 及类似系统）

dpkg -i package.deb 安装/更新一个 deb 包

dpkg -r package_name 从系统删除一个 deb 包

dpkg -l 显示系统中所有已经安装的 deb 包

dpkg -l | grep httpd 显示所有名称中包含“httpd”字样的 deb 包

dpkg -s package_name 获得已经安装在系统中一个特殊包的信息

dpkg -L package_name 显示系统中已经安装的一个 deb 包所提供的文件列表

dpkg --contents package.deb 显示尚未安装的一个包所提供的文件列表

dpkg -S /bin/ping 确认所给的文件由哪个 deb 包提供

APT 软件工具（ Debian,Ubuntu 及类似系统）

apt-get intall package_name 安装/更新一个 deb 包

apt-cdrom install package_name 从光盘安装/更新一个 deb 包

apt-get update 升级列表中的软件包

apt-get upgrade 升级所有已安装的软件

apt-get remove package_name 从系统删除一个 deb 包

apt-get check 确认依赖的软件仓库正确

apt-get clean 从下载的软件包中清理缓存

apt-cache search searmhed-package 返回包含所要搜索字符串的软件包名称

查看文件内容

cat file1 从第个字节开始正向查看文件的内容

tac file1 从最后一行开始反向查看一个文件内容

more file1 查看一个文件的内容

less file1 类似与‘more’命令，但时它允许在文件中和正向操作一样的反向操作

head -2 file1 查看一个文件的前两行

tail -2 file1 查看一个文件的最后两行

tail -f /var/log/messages 实时查看被添加到一个文件中大的内容

文本处理

cat file1 file2 ... |command <>file1_in.txt_or_file1_out.txt general syntax for text maniprlation using PIPE,STDIN and STDOUT

cat file1 | command(sed, grep, awk, grep, et...) >result.txt 合并一个文件的详细说明文本，并将简介写入一个新文件中

cat file1 | command(sed, grep, awk, grep, etc...) >>resulttxt 合并一个文件的详细说明文本，并将简介写入一个已有的文件夹中

grep Aug /var/log/messages 在文件‘/var/log/massages’中查找关键词“Aug”

grep ^Aug /var/log/messages 在文件‘/var/log/messages’中查找以“Aug”开始的词汇

grep [0-9] /var/og/messages 选择‘/var/log/messages’文件中所包含数字的行

grep Aug -R /var/log/* 在目录‘/var/log’及随后的目录中搜索字符串“Aug”

sed 's/stringa1/stringa2/g' example.txt 将 example.txt 文件中的“string1”替换成“string2”

sed '/^$/d' example.txt 从 example.txt 文件中删除所有空白行

sed '/*#/d; /^$/d' example.txt 从 example.txt 文件中删除所有注释和空白行

echo 'esempio' |tr '[:lower:]' '[:upper:]' 合并上下单元格内容

sed -e 'ld' result.txt 从文件 example.txt 中排除第一行

sed -n '/stringal/p' 查看只包含词汇“string1”的行

sed -e 's/*$//' example.txt 删除每一行最后的看空白字符

sed -e 's/stringal//g' example.tst 从文档中只删除词汇“stringl”并保留剩余全部

sed -n '1,5p;5q' example.txt 查看从第一行到第 5 行内容

sed -n '5p;5q' example.txt 查看第 5 行

sed -e 's/00*/0/g' example.txt 用单个零替换多个零

cat -n fil

e1 表示文件的行数

cat example.txt | awk 'NR%2==1' 删除 example.txt 文件中的所有偶数行

echo a b c | awk'{ print $1}' 查看一行第一栏

echo a b c | awk ' {print $1,$3}' 查看一行的第一和第三栏

paste file1 file2 合并两个文件或两栏的内容，中间用“+”区分

sort file1 file2 排序两个文件的内容

sort file1 file2 | uniq 取出两个文件的并集（重复的行只保留一份）

sort file1 file2 |uniq -u 删除交集，留下其他的行

sort file1 file2 |uniq -d 取出两个文件的交集（只留下同时存在于两个文件中的文件）

comm -1 file1 file2 比较两个文件的内容只删除 ‘file1’ 所包含的内容

comm -2 file1 file2 比较两个文件的内容只删除 ‘file2’ 所包含的内容

comm -3 file1 file2 比较两个文件的内容只删除两个文件共有的部分

字符设置和文件格式转换

dos2unix filedos.txt fileunix.txt 将一个文本文件的格式从 MSDOS 转换成 UNIX

unix2dos fileunix.txt filedos.txt 将一个文本文件的格式从 UNIX 转换成 MSDOS

recode ..HTML<page.tt >page.html 将一个文本文件转换成 html

recode -l |more 显示所有允许的转换格式

文件系统分析

badblocks -v /dev/hdal 检查磁盘 hdal 上的坏次块

fsck /dev/hdal 修复/检查 hdal 磁盘上 linux 文件系统的完整性

fsck.ext2 /dev/hdal 修复/检查 hdal 磁盘上 ext2 文件系统的完整性

e2fsck /dev/hdal 修复/检查 hdal 磁盘上的 ext2 文件系统的完整性

e2fsck -j /dev/hdal 修复/检查 hdal 磁盘上 ext3 文件系统的完整性

fsck ext3 /dev/hkal 修复/检查 hdal 磁盘上 ext3 文件系统的完整性

fsck .vfat /dev/hdal 修复/检查 hdal 磁盘上 fat 文件系统的完整性

fscm .msdos /dev/hdal 修复/检查 hdal 磁盘上 dos 文件系统的完整性

dosfsck /dev/hdal 修复/检查 hdal 磁盘上 dos 文件系统的完整性

初始化一个文件系统

mkfs /dev/hdal 在 hdal 分区创建一个文件系统

mke2fs /dev/hdal 在分区创建一个 linux ext2 的文件系统

mke2fs -j /dev /hdal 在 hdal 分区创建一个 linux ext3(日志型) 的文件系统

mkfs -t vfat 32 -F /dev/hdal 创建一个 FAT32 文件系统

fdformat -n /dev/fd0 格式化一个软盘

mkswap /dev/hda3 创建一个 swap 文件

SWAP 文件系统

mkswap /dev/hda3 创建一个 swap 文件系统

swapon /dev/hda3 启用一个新的 swap 文件系统

swapon /dev/hda2 /dev/hdb3 启用两个 swap 分区

备份

dump -0aj -f /tmp/home0.bak /home 制作一个 ‘/home’ 目录的完整备份

dump -1aj -f /tmp/home0.bak /home 制作一个 ‘/home’ 目录的交互式备份

restore -if /tmp/home0.bak 还原一个交互式备份

rsync -rogpav --delete /home /tmp 同步两边的目录

rsync -rogpav -e ssh --delete /home ip_address:/tmp 通过 SSH 通道 rsync

rsync -az -e ssh --delete ip_addr:/home/public /home/local 通过 ssh 和压缩将一个远程目录同步到本地目录

rsync -az -e ssh --delete /home/local ip_addr:/home/publi

c 通过 ssh 和压缩将本地目录同步到远程目录

dd bs=1M if=/dev/hda | gzip | ssh user@ip_addr 'dd of+hda.gz' 通过 ssh 在远程主机上执行一次备份本地磁盘的操作

dd if=/dev/sda of=/tmp/file1 备份磁盘内容到一个文件

tar -Puf dackup.tar /home/user 执行一次对 ‘/home/user'目录的交互式备份操作

(cd /tmp/local/&& tar c.) | ssh -C user@ip_addr 'cd /home/share/ && tar x -p' 通过 ssh 在远程目录中复制一个目录内容

(tar c /home) | ssh -C user@ip_addr ' cd /home/backup-home && tar x -p' 通过 ssh 在远程目录中复制一个本地目录

tar vf - . | (cd /tmp/backup ; tar xf -) 本地将一个目录复制到另一个地方，保留原有权限及链接

find /home/user1 -name '*.txt' | xargs cp -av --target-directory=home/backup/

--parents 从一个目录查找并复制所有以‘.txt’结尾的文件到另一个目录

find /var/log -name '*.log' |tar cv --files-from=- \ bzip2 > log.tar.bz2 查找所有以‘.log’结尾的文件并做成一个 bzip 包

dd if=/dev/hda of=/dev/fd0 bs=512 count=1 做一个将 MBR(Master Boot Record)内容复制到软盘的动作

大的 i=/dev/fd0 of=/dev/hda bs=512 count=1 从已经保存到软盘的备份中恢复 MBR 内容

光盘

cdrecord -v gracetime=2 dev=/dev/cdrom -eject lank=fast -force 请看空一个可复写的光盘内容

mkisofs /dev/cdrom >cd.iso 再磁盘上创建一个光盘的 iso 镜像文件

mkisofs /dev/cdrom | gzip >cd_iso.gz 在磁盘上创建一个压缩了的光盘 iso 镜像文件

mkisofs -J -allow-leading-dots -R -V"Label CD" -iso-level 4 -o ./cd.iso data_cd 创建一个目录的 iso 镜像文件

cdrecord -v dev=/dev/cdrom cd.iso 刻录一个 ISO 镜像文件

gzip -dc cd_iso.gz | cdrecord dev=/dev/cdrom - 刻录一个压缩了的 ISO 镜像文件

mount -o loop cd.iso /mnt/iso 挂载一个 ISO 镜像文件

cd-paramia -B 从一个 CD 光盘转录音轨到 wav 文件中（参数-3）

cdrecord --scanbus 扫描总线以识别 scsi 通道

dd if=/dev/hdc |md5sum 校验一个设备的 md5sum 编码，例如一张 CD

应用命令

whois 域名/ip 查看域名的详细信息

ping 域名/ip 测试本机到远端主机是否联通

dig 域名/ip 查看域名解析的详细信息

host -l 域名 dns 服务器传输 zone

扫描

nmap:

-sS 半开扫描 TCP 和 SYN 扫描

-sT 完全 TCP 链接扫描

-sU UDP 扫描

-PS sym 包探测（防火墙探测）

-PA ack 包探测 （防火墙探测）

-PN 不 ping

-n 不 dns 解析

-A -O 和-sV

-O 操作系统识别

-sV 服务器版本信息(banner)

-P 端口扫描

-T 设置时间级别（0-5）

-iL 导入扫描将结果

-oG 输出扫描结果

操作系统识别：

p0f -i eth0 -U -p 开启混杂模式

xprobe2 ip|域名 检测 os

banner 获取：

nc ip port 检测端口是否打开

telnet ip port 检测端口是否打开

wget ip 下载主页

cat index.html | more 是否显示主页代码

q 推出

windows 枚举

nmap -sS -p 139,445 ip 扫描 windows

cd /pente

st/enumeration/smb-enum

nbtscan -f targetIP 检测 netbios

smbgetserverinfo -i targetIP 扫描 name,os,组

smbdumpusers -i targetIP 列出用户

smbclient -L //targetIP 列出共性

使用 windows：

net use \\ipipc$"" /u:"" 开启看空会话

net view \\ip 显示共享信息

smbclient:

smbclient -L hostName -I targetIP 枚举共享

smbclient -L hostName/share -U "" 用看空用户链接

smbclient -L hostName -I targetIP -u admin 普通用户链接

rpcclient:

rpcclient targetIP -u "" 打开一个空会话

netshareenum 枚举共享

enumdomusers 枚举用户

lsaenumsid 枚举域 SID

queryuser RID 查询用户信息

createdomuser 创建用户访问

ARP 欺骗：

ettercap:

nano /usr/local/etc/etter.conf 配置文件

Sniff >Unified sniffing >Network interface: eth0 >OK 设置抓包网卡

Hosts >Scan for hosts (do this two times) 扫描网段的主机

Hosts >Hosts list 显示主机列表

Select the default gateway >Add to Target 1 添加主机

Select the target >Add to Target 2 添加主机

Mitm >Arp poisoning >Sniff remote connections >OK 设置 AEP 攻击

start >Start sniffing 开始攻击

dsniff -i eth0 监听网卡窃喜登录用户密码

urlsnarf -i eth0 嗅探 http 请求

msgsnarf -i eth0 嗅探聊天软件的聊天内容

driftnet -i eth0 网络管理嗅探图片，音频

dns 欺骗

nano /usr/local/share/ettercap/etter.dns 编辑配置文件

Plugins >Manage the plugins >dns_spoof 设置 dns 欺骗

Mitm >Arp poisoning >Sniff remote connections >K 设置 RP

Start >Start sniffing 开始攻击

Exploits 漏洞利用：

cd /pentest/exploits/exploit-db 进入目录

cat sploitlist.txt | grep -i[exploit] 查询需要的漏洞

cat exploit | grep "#include " 检查运行环境

cat sploitlist.txt | grep -i exploit | cut -d " " -f1 | xargs grep sys | cut -d ":" -f1 | sort -u

只保留可以在 linux 下运行的代码

Metasploit:

svn update 升级

./msfweb Web 接口 127.0.0.1：55555

./msfconsole 字符下的 Console

help 帮助

show

显示选项

search 搜索名字

use 受用漏洞

show options 显示选项

Set 设置选项

show payloads 显示装置

set PAYLOAD 设置装置

show options 显示选项

set

show targets 显示目标（os 版本）

set TARGET 设置目标版本

exploit 开始漏洞攻击

sessions -l 列出会话

sessions -i 选择会话

sessions -k 结束会话

z 把会话放到后台

c 结束会话

jobs 列出漏洞运行工作

jobs -k 结束一个漏洞运行工作

show auxiliary 显示辅助模块

use 使用辅助模块

Set 设置选项

run 运行模块

scanner/smb/version 扫描系统版本

scanner/mssql/mssql_ping 测试 mssql 是否在线

scanner/mssql/mssql_login 测试登录（暴力或字典）

Attacker behind firewall: bind shell 正向

Target behind firewall: reverse shell 反向

Meterpreter 衔接不懂 dos 可以用这个：

db_import_nessus_nbe 加载 nessus 的扫描结果

db_import_nmap_xml 加载 nmap

的扫描结果

自动化攻击流程：

cd /pentest/exploit/framework3

./msfconsole

load db_sqlite3

db_destroy pentest

db_create pentest

db_nmap targetIP