centos 6 RHEL6 vmware

vmware

客户端操作系统:
  要安装系统相同 RHEL6

  OURLAB

  虚拟机为系统分配内存(不同的系统会有不同分配方案) rhel6分配4G内存,不会立即使用物理内存中的4G

  虚拟机:上网 建设服务器

  磁盘:虚拟机中 文件形式存在(镜像文件 image file)

  存储为单个文件:NTFS FAT32
  拆分为多个文件:

  安装语言:

  主机名:www.guan.com

  网卡:

  时区:亚洲/上海 Asia/Shanghai

  root密码:root是Linux系统管理员用户

  "nihao12!"

  挂起:暂停(保持在某一个状态)

  快照:保存某一个状态 LANMP http

ifconfig:查看设置LinuxIP地址

宿主机<--->虚拟机 通讯

虚拟机:桥接模式(IP地址:10.0.0.0/24) 宿主机:物理网卡(10.0.0.4/24)

虚拟机:NAT模式(网络地址转换) 宿主机:VMware Network Adapter VMnet8

虚拟机:仅主机 宿主机:VMware Network Adapter VMnet1

操作系统概念:

计算机体系结构 linux由来 linux设计思想

体系结构:

计算器:cpu

控制器:cpu

存储器:内部存储(内存) 寻址空间 (MP3) 国家:每个国家做成一个单元 某个国家:河北省 北京市 天津市 北京市:每个区

输入设备:鼠标,键盘

输出设备:显示器 音频等

操作系统:带有很多外围程序的系统 笔记本硬件--win7(计算器/画图工具)

office:word ppt 表格等 打印功能

系统调用:system call

打印功能:小程序

直接作用到硬件上

直接作用到内核上:效率高

API:application program interface 应用程序接口

web
qq
office

随系统一起启动

不随系统一起启动应用程序:交互式应用程序(用户使用时根据需要启动该程序)

操作系统:微软公司 Linux--unix

1969: MIT BELL:MULTICS(汇编语言)
ken:小游戏(雷霆战机 太空遨游)

ken:PDP-11 PDP-7

C语言重新开发MULTICS
unics---unix
1977:BSD:openbsd freebsd netbsd

1984年:谭宁邦教授 minix mini unix:为linux出现产生重大的影响,广泛应用到教学中

1984年:GNU: GNU is not unix (GCC/BASH SHELL)史托曼教授:自己开发的软件让别人使用
GPL:
取得软件和源码:任何人可以根据自己的需求自由的获取软件和源码
复制:复制该软件(衍生)
修改:改源码可以自行修改(衍生)
在发行:把修改后的代码进行二次发行
修改授权:
销售:修改后的软件(获取的软件)进行商业销售
LGPL:基于某个内核平台开发的软件可以进行销售

qq:腾讯公司 windows

qq游戏(小游戏)

芬兰大学生 :托瓦兹得到源码 unix-like

x386:

源码放到FTP上:没有XX硬件驱动

虚拟团队诞生: 1991年linux系统第一个版本诞生 linux 0.0.2

linux版本号(内核版本号):三部分组成:

A.B.C

A:主版本号

B:次版本号(奇数:开发版本 偶数:发行版本)

C:修订版本号(修订次数)

内核版本:3.6.28 3时主版本号 6:次版本号 28:修订了28次

先把3.6.28拷贝一份进行研发,并把版本号升级为3.7.01(在3.6.28基础上修复漏洞,开发新功能)

3.8.XX:下一个发行版本

开发--内测(修复漏洞,开发新功能)--公测--正式版本生成(正式上线)

[root@www ~]# 管理员登录

[zhangsan@www ~]$ 普通用户登录

root:当前登录的用户名

www:主机名

~:当前用户所处的目录(~代表用户宿主目录)

#:代表当前用户是管理员

$:代表当前登录用户是普通用户

linux设置哲学思想:

远程连接linux系统:xshell使用 CRT

IP地址:ifconfig查看IP地址

xshell: ssh 192.168.10.147

shell的作用

shell:使用者,计算机交互接口(人机交互接口)

shell中输入命令,shell把命令传递给内核,内核把命令结果反馈给shell,人从shell中读取命令执行结果

内核:图形界面/字符界面

linux图形界面:

GNOME:linux默认图形界面,C语言开发

KDE:C++开发

xface:简化的图形界面

linux支持的shell:

bash:linux默认支持的shell

ksh

csh

内核的作用:

进程管理:进程:要执行的任务

内存管理:

文件系统:存储设备上存储数据的方式方法:windows:NTFS/FAT32 LINUX:ext3 ext4 xfs等

网络功能:管理IP地址信息等

硬件驱动:

安全功能:

linux轻巧,稳定系统

linux设计思想

1:有很多的小程序组成,每个小程序完成单一的功能,实现复杂的任务(http服务需要安装很多小组件)

2:一切皆文件:所有的外围设备(硬件)或者其他程序

3:尽量避免捕获用户接口

4:配置文件保存为纯文本格式(可以用文本编辑器编辑 vim)

Linux终端:多用户多任务系统

6个终端:ctrl+alt+F1-F6

Linux界面:

GUI:graphical user interface:图形用户接口(图形界面) 切换图形界面:ctrl+alt+F7

CLI: command line interface:命令行接口(字符界面)

命令格式等内容

命令提示符(prompt): [root@www ~] #

退出当前终端:exit

Linux使用凭证: 指纹/刷脸等 用户名和密码(用户获取资源权限的凭证)

root:linux默认管理员用户名

普通用户:

切换用户:su(switch user)

su命令:当root用户切换到普通用户时,不需要普通用户的密码;而当普通用户切换到root用户时,需要提供root用户密码

root student

su user_name

$ exit 退出当前用户

Linux命令格式:

命令字 [选项] [参数]:中括号表示可以省略

命令字:唯一的,实现某一项功能

选项:修改命令的执行方式(实现特定功能)

长选项:--引导,是一个单词(多个长选项不能组合)

短选项:-引导,是一个字符(多个短选项可以组合)

参数:命令作用的对象

内核版本号:主.次.修订 3.6.28

RHEL 6 7(系统的发行版本号)

系统中:用户名 UID
root: 密码
student:

3A认证:
认证机制:authentication
密码认证:
1:符合复杂性要求(数字,小写字母,大写字母,特殊符号至少三种)
2:密码长度(至少7位)
3:不要使用易记得密码(使用随机字符)
4:定期更改密码
5:重复密码的时间要长
授权机制:authorization
审计机制:audition(日志)

Linux登录信息:/etc/issue

Linux命令分类:
内部命令:shell自带的命令
外部命令:在Linux文件系统中存在一个应用程序

type:查看Linux命令类型
[command] is a shell builtin:buildin关键字说明该命令是内部命令
mkdir is /bin/mkdir:有路径显示(外部命令)

路径:
绝对路径:从根(/)开始的路径是绝对路径,linux系统中只有一个根
相对路径:以所处的工作目录为参照点（.代表当前路径 ..代表上一级路径 ~代表登录用户的宿主目录)

ls(list):列出目录
-l(--long):以长格式显示

-rw-------. 1 root root 1207 Nov 6 18:51 anaconda-ks.cfg

第一位:文件类型

-:代表普通文件(file)

d:代表目录(directory)

b:块设备(block):如硬盘,U盘等

c:字符设备(char):如键盘等

s:套接字文件(socket)

p:命名管道(pipe)

l:符号链接文件(symbolic link file)

第二到十位:权限位(rwx:读写执行)

234位:文件属主权限(owner)

567位:文件属组权限(group)

8910位:其他用户权限(other)

1:代表文件硬链接的次数

第一个root:代表文件属主(owner)

第二个root:代表文件属组(group)

1207:文件的大小

Nov 6 18:51:文件最后被访问的时间戳(stat命令查看文件时间戳)

文件的时间戳:

access:访问的时间戳

Modify:文件被修改时间戳(修改文件数据:添加删除数据等)

change:文件被更改时间戳(更改文件属性)

anaconda-ks.cfg:文件名

-h:为文件大小添加单位

-a:显示目录下所有文件(包括. .. 以.开始的隐藏文件)

-A:和-a相同,但是不显示.和..

-R:递归显示目录中的内容(一并显示子目录中所有内容)

-r:逆序显示目录内容

-i:显示文件所在的inode节点(index node)

-d:显示目录本身的属性

环境变量:内存中的命名空间

PATH变量:存放系统命令路径,以冒号隔开

查看PATH变量:echo $PATH(Linux命令严格区分大小写)

[root@www ~]# echo $PATH
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin

如果在以上路径中没有找到相关命令(该命令为外部命令),提示用户command not found

如果一个命令在以上多个路径中存在,系统会按照从前往后的顺序查找,查找到该命令后,后面的路径的不在查找

hash:查看命令缓存及命中率

Linux命令的帮助信息:

内部命令:help [command]

外部命令:[command] --help

man:帮助(命令的使用说明书)(查看内部命令时,显示的是bash帮助信息)

语法: man [command]

上下方向键:翻行

enter:向下翻行

pagedown:向下翻页

pageup:向上翻页

/word:从上往下查找关键字

?word:从下往上查找关键字

q:退出当前帮助信息

man目录说明

1:User Commands:普通用户命令(/bin /usr/bin /usr/local/bin):binary二进制

2:System Calls:系统调用库

3:C Library Functions:库调用

4:Devices and Special Files:设备或特殊文件(硬件设备)(硬件设备存放在/dev)

5:File Formats and Conventions:查看配置文件格式

6:Games et. Al.:游戏

7:Miscellanea:杂项

8:System Administration tools and Deamons:管理命令(/sbin /usr/sbin /usr/local/sbin)sbin:secret binary

man使用说明:

NAME:命令名称及简要用法

SYNOPSIS:语法格式,可能包括一些选项的使用

DESCRIPTION:命令和命令选项的详细说明

Exit status:退出状态码

AUTHOR:作者信息

REPORTING BUGS(BUG):发现BUG时如何反馈信息

COPYRIGHT:该命令的版权信息

SEE ALSO:另外参照的帮助信息

OPTIONS:说明该命令每一个选项的详细用法

EXAMPLES:命令使用实例

man手册页目录:/usr/share/doc(工作中主要是查看内核信息)

<>:必须使用的选项或参数,不可以省略

[]:可是省略的选项或参数

...:可以使用多个选项或参数

|:代表多选一

{}:分组,没有特殊意义

info:在线查看帮助,注重于命令历史,版权信息等

cd:change directory(切换目录)

cd:不加选项,返回到当前用户的宿主目录

-:返回到上一次的工作路径

~username:切换到username的宿主目录

pwd:print working directory:显示当前所处的工作目录

which:查看命令所在的路径

whatis:查看命令所在的帮助信息目录,该命令在系统启动大约70分钟后会生成whatis的数据库,如果时间太短,该命令不会执行,我们可以使用makewhatis(CentOS 6)初始化该命

令,在CentOS 7中使用mandb命令,

yum -y install man-pages

Linux根文件系统:

根目录下目录及子目录的作用

Linux文件类型:

-:普通文件

纯文本文件(ASCII):配置文件

二进制文件(binary file): 命令

数据格式文件(data):/var/log/wtmp

d:目录文件

l:链接文件:软链接文件

设备文件:(/dev)

b:block块设备

c:character字符集设备:一次性读取,按顺序读取

s:socket套接字文件:通常用在网络上数据连接: IP:PORT

p:管道:特殊的文件类型,解决多个程序同时访问一个文件所造成的错误问题

file:查看linux中文件类型

语法: file [options] file_name

linux文件名限制:

1:单个文件或目录的名称不能超过255字符

2:文件命中不能包含特殊字符(/ . - +等)

FHS标准(filesystem hierarchy standard):希望用户可以了解已经安装的程序在哪个目录下

根文件文件系统(/):rootfs:root filesystem

/bin:二进制,存放命令

/boot:存放启动和内核相关文件

/dev:存放设备文件

/etc:存放应用程序的配置文件

/home:普通用户的家目录,默认为/home/USERNAME

/lib和/lib64:存放系统开机时需要用的函数库及/bin和/sbin命令调用函数库

/lib/modules:存放内核相关的模块(驱动程序等)

/media和/mnt:挂载点,/media挂载移动设备 /mnt挂载临时设备

/opt:第三方软件存放目录(用户自行安装的软件存放处),现在一般安装到/usr/local下

/proc:伪文件系统,数据存放在内存中,存放关于进程的相关信息

/root:管理员的家目录

/sbin:存放管理员使用的命令

/srv:service缩写,存放服务数据目录,如可以把www服务的网页存放到该目录

/tmp:存放临时文件,所有用户都可以访问创建文件,但是每个用户只能删除自己的文件

/sys:伪文件系统,存放在内存中,记录内核相关的信息,包括目前加载内核模块和内核检测到的硬件设备等

/usr:UNIX software resource:存放安装的应用程序

/usr/bin:普通用户使用的命令(和/bin区别是否与开机有关)

/usr/sbin:网络服务器命令

/usr/lib和/usr/lib64:包含各种应用程序函数库

/usr/share:存放共享文件目录(在线帮助文件,杂项,时区文件等)

/usr/include:存放头文件

/usr/src:释放源代码目录

/var:vary缩写,存放经常变动的文件,比如日志,mail等

/var/cache:存放应用程序运行时产生的缓存文件

/var/lib:程序运行时,需要使用的数据文件的存放目录

/var/lock:设备或资源一次只能被一个应用程序使用,如果多应用程序会产生错误,因为要为设备或资源上锁(存放锁文件)

/var/log:存放日志目录(系统,用户登录,服务日志等)

/var/mail:存放个人电子邮件(系统报警产生邮件信息等)

/var/run:存放应用程序运行时PID文件(进程号.pid结尾)

/var/spool:存放队列数据,排队等待其他用户程序使用的数据,数据通常使用完成后会被删除

FHS规定:/etc /bin /dev /lib /sbin五个目录必须要和根目录位于同一文件系统

基本命令:ls cd pwd

学习内容:file cat cp mv rm mkdir touch tree which whereis

文件管理
目录管理
系统管理
网络管理等

file:查看文件类型(windows是用扩展名识别文件类型)
语法:file [options] [args]
-b:显示结果时,不显示文件名
-c:显示执行file命令的执行过程(file是如何去判断文件类型),便于排错或分析file命令执行过程
-i:输出MIME类型的字符串
-z:显示压缩文件的内容
-L:查看软链接对应文件的类型
-f:查看文件中文件名的类型
cat:1:一次性查看整个文件
语法:cat [options] [args]
2:从键盘输入创建一个新文件或向现有文件中添加新数据:
新建新文件:cat >new_file_name <<EOF //EOF:end of file
>Linux command //从键盘输入
>Uinx command //从键盘输入
>EOF //从键盘输入,以结束书写
向现有文件中追加数据:
cat >> file_name << EOF
>CISP!
>CISSP!
>EOF
3:把多个文件内容合并到一个文件输出:
cat file1 file2 > file3

cat /etc/passwd /etc/shadow > /root/user.txt

-n:显示文件内容时同时显示行号,包括空行
-b:和-n功能相同,但不包括空行
-S:当文件中有多个空行时,合并为一个空行
-E:在显示内容时,结尾添加$符号

cp:复制文件或目录(copy简写)

语法:cp [options] [src_file] [des_file]

-f:force,强制复制文件或目录不进行提示

-r:递归复制目录

-s:为某个文件创建符号链接(软链接),而不是复制文件

-b:覆盖已有的文件前,对目标文件进行备份

-l:为文件创建硬链接,而不是复制文件

-p:复制文件时保留文件的原有属性

-d:当复制软链接文件时,把目标文件或目录也会创建为软链接,并指向最原始的文件

-i:覆盖目标文件前询问(cp = cp -i)

mv:move,移动或从命名文件和目录(当原位置和目标位置是同一目录时,是重命名;当原位置和目标位置不是同一目录时,是移动)

-b:移动前先对源文件进行备份

-f:强制覆盖

-i:覆盖目标文件前询问

-t:将多个文件移动同一个目录(目标目录在前面,后面跟文件 mv -t [directory] file1 file2 ...)

rm:remove,删除文件或目录

-d:删除可能存在数据的目录

-f:强制删除

-i:删除前进行询问

-r:递归删除整个目录

-v:显示命令执行过程

mkdir:make directory,创建空目录

-m:创建目录时同时设置权限

-p:递归创建新目录

-v:显示创建目录的过程

touch:创建新文件或者修改文件时间戳

-a:只改变访问时间

-c:不创建文件

-d:使用指定时间戳创建新文件,而不是使用系统时间
-t:使用指定格式时间戳创建新文件,而不是使用系统时间
-f:解决与BSD系列Unix系统兼容性
-m:只更改变动时间

tree:查看目录树(tree = ls -R)

which:查找命令所在的路径

whereis:查看命令所在的路径,源代码文件,帮助信息文件所在的文件
-b:只查看文件所在的位置(等同于which命令)
-m:只查看帮助信息所在的位置
-s:支持看源代码所在的位置

查找文件命令:locate
非实时性,根据自己的数据库查找文件,CentOS每天更新一次数据库,安装完locate后,生成/var/lib/mlocate目录,存放数据库
模糊匹配
/var/lib/mlocate/mlocate.db //locate工具的数据库
yum -y install mlocate //安装locate工具
locate file_name
updatedb //更新locate数据库

查看文本文件内容命令:cat more less head tail
Linux和Windows中文件内容行尾标记是不同的:
Linux中是以$为结尾
Windows中是以ENTER键结尾

cat:一次性查看文件内容

more:可以翻页查看文本文件内容
enter键:向下翻一行
空格键:向下翻一屏
ctrl+b:向上翻一屏
ctrl+f:向下翻一屏

=:显示当前的行号
v:调用vi编辑器
!command:调用shell执行命令
q:退出more命令

more +num file_name //从num行开始查看文件内容

less:可以翻页查看文本文件内容
enter键:向下翻一行
空格键:向下翻一屏
pagedown:向下翻一屏
pageup:向上翻一屏
上下方向键:向上/向下翻一行
/word:在该文件中查找word关键字
n:查找下一个
N:查找上一个

管道:连接多条命令,前一条命令输出结果作为后一条命令的输入条件, 管道符 |

head:默认查看文件文件的前十行
-n:查看文件文件前n行

tail:默认查看文本文件的后十行
-n:查看文本文件的后n行
-f:动态查看文本文件内容,如动态查看日志: tail -f /var/log/messages

shell广义上分为两类:
GUI:包括GNOME KDE XFACE等
CLI:sh csh ksh bash等(Linux发行版本中,bash是默认使用的shell程序)

shell启动:当用户登录完成后,系统会自动启动shell程序
进程:应用程序的副本;使用PID区分(在系统中,一个进程只认为自己存在)
root student用户
shell:父shell程序和子shell程序(父子shell之间设置环境互相独立)

cat /etc/shells //查看当前系统所支持的shell程序
yum -y install ksh csh //安装ksh,csh的shell程序
exit //退出当前shell程序
 shell是一个程序

bash的特性:
 1:命令历史:Linux会自动记录系统过去执行的命令,并保存在内存的缓冲区中
  在每个用户的家目录下,有个隐藏文件.bash_history保存命令历史
 # history //查看linux的历史
  -c:清空命令历史
   -d:删除某一条命令历史
   -w:将命令历史保存到某个文件中
 变量:PATH命令路径变量
   HISTSIZE:命令历史大小变量
   $ echo $HISTSIZE //查看HISTSIZE变量的值,默认的命令历史是1000条
 命令历史使用技巧:
  !n:执行命令历史中的第n条命令(n是命令历史编号)
   !-n:执行命令历史中的倒数第n条命令(n是命令历史编号)
   !word:执行命令历史中最近一次以word开始的命令(word必须能够唯一的标识用户想执行的命令)
   !!:执行上一条命令
   !$:引用一个命令的最后一次参数
   ESC键(按完松开) 再按.键:引用一个命令的最后一次参数

 2:管道,IO重定向
  计算机体系结构:
   控制器:CPU,读取系统指令
   运算器:CPU,运算
    存储器:RAM(内部存储器:易失性存储器)
    输入设备(Input):输入数据,如键盘,硬盘等
    输出设备(Output):显示指令执行结果,如显示器,音响,硬盘等

   计算机总线:
    地址总线:负责内存寻址

数据总线:负责传输数据

控制总线:负责控制指令

寄存器:CPU中内部临时存储空间

I/O设备:负责计算机内部存储设备和外部存储设备(如硬盘,光盘,U盘等)进行交互的设备

程序:指令+数据

指令:有程序提供,负责加工数据

数据:系统中数据可以有多种来源,比如来自变量,来自文件,来自输入设备等

当用户没有为指令指定数据来源时,系统要有默认的数据来源

标准输入输出设备:

标准输入设备:键盘(stdin),文件描述符为0

标准输出设备:显示器(stdout),文件描述符为1

标准错误输出设备:显示器(stderr),文件描述符为2

三种数据流:

标准输入数据流/标准输出数据流/标准错误输出数据流

当在Liunx中打开一个文件时,内核会反复调用,对于文件标识就很重要了,用文件描述符来标识文件,文件加载完成用数字标识

fd:file descriptor(文件描述符)

IO重定向:把默认输入输出数据来源,重新定向到其他的文件或设备

输出重定向:

>:覆盖输出重定向

>>:追加输出重定向

2>:错误覆盖输出重定向

2>>:错误追加输出重定向

&>:混合覆盖输出重定向

&>>:混合追加输出重定向

输入重定向:

<:输入重定向

管道:连接多条命令,把前一条命令的输出结果作为后一条命令的输入条件(组合小程序,实现大功能)

command1 | command2 | commad3 ....

tr:实现字符转换,不修改源文件,语法:tr [OPTION]... SET1 [SET2]

tee:

3:命令别名(alias定义的别名只在当前shell生效)

alias COMM_ALIAS=COMMAND //只执行alias命令,列出系统中所有的命令别名

注意:执行alias命令时,COMMAND最好用''(引号)引起来

unalias COMM_ALIAS //取消命令别名

4:命令行编辑

ctrl+a:光标快速跳转到命令行的行首

ctrl+e:光标快速跳转到命令行的行尾

ctrl+u:快速删除光标位置到命令行行首的字符

ctrl+k:快速删除光标位置到命令行行尾的字符

ctrl+l:清屏(命令clear:清屏)

5:命令行展开

5.1:命令补齐:在PATH变量搜索命令并补齐(PATH变量必须正常,输入要补齐命令的字符数一定能够唯一标识这条命令;tab键补齐)

按tab键两次,列出以某个字符开始的所有的命令

echo $PATH //查看PATH变量的值

路径和文件补齐:在系统路径中查找

命令行补齐功能不能补齐选项

5.2:命令替换(经常用到shell脚本编程中) $(command)或者 `command`(两边是反撇号)

命令替换就是把命令中的子命令替换成子命令执行结果的过程

echo "字符串"

""(双引号):弱引用(可是实现变量的替换,把变量名替换为变量值)

``(反撇号):命令引用

''(单引号):强引用(不能完成变量替换)

6:命令行通配:globbing

*:匹配任意长度的任意字符

?:匹配任意单个字符

[]:匹配指定范围内的任意单个字符

[ab] [a-m] [a-z] [A-Z] [a-zA-Z] [0-9] [a-zA-Z0-9]

[^]:匹配指定范围外的任意单个字符

[^a-z] [^0-9]

[:space:]:表示空格

[:punct:]:表示所有标点符号

[:lower:]:表示所有小写字母

[:upper:]:表示所有大写字母

[:alpha:]:表示所有字母(包括大小写)

[:digit:]:表示所有数字

[:alnum:]:表示所有数字和大小写字母

7:变量

8:编程

硬盘:OS--应用程序

process(进程):用户执行应用程序的过程

权限:用户对于应用程序的执行能力

文件属主:

属组:

其他用户:

用户:多任务,多用户

Linux实现权限隔离机制(多用户)

操作系统识别用户:

用户:UID

组:GID(逻辑容器:包含用户;实现多个用户对于某个文件或应用程序分配相同的权限)

用户分类:

管理员用户:root UID:0

普通用户:1000-65535

程序(系统)用户:1-999

组分类:

管理员组/普通组

基本组:Linux中,新建用户时,没有指定隶属于的组,系统会为该用户建立一个同名组,作为该用户基本组

附加组:方便以后为用户分配权限

解析:

/etc/passwd:存储用户基本信息

/etc/shadow:存储用户的影子口令

/etc/group:存放组的基本信息

/etc/passwd解释:用冒号隔开

第一列:用户的登录名

第二列:加密密码(x表示密码占位符,密码保存在/etc/shadow中)

第三列:用户的ID号

第四列:组的ID号

第五列:用户的描述信息

第六列:用户宿主目录

第七列:用户默认使用的shell(/etc/shells)

/etc/shadow解释:用冒号隔开

第一列:用户的登录名

第二列:用户加密后的密码

第三列:最后一次更改密码时间(从1970年1月1号)

第四列:密码最小使用时间

第五列:密码最长使用时间

/etc/shadow密码区域解释(格式:idsalt$encrypted):

三部分组成:用$隔开

第一部分加密算法:

1:MD5加密

5:SHA-256加密

6:SHA-512加密

第二部分:随机序列号

第三部分:随机序列号和密码共同加密后的字符串

加密算法:

对称加密:加密和解密使用相同的密码(效率高)

非对称加密:加密和解密使用不同的秘钥,公钥和私钥

单向加密,散列加密:提取数据的特征码,常用于数据完整性校验

1:不可逆

2:定长输出

md5: 128位定长输出

sha: 160位 224位 256 位 384位 512位

3:雪崩效应:输入数据一小点儿的变化,会引起结果的巨大变化

新建用户命令:useradd

/etc/default/useradd文件解释(只对新建用户生效):

useradd defaults file //注释行

GROUP=100 //可以创建普通组

HOME=/home //普通用户宿主目录位置(在/home下生成和用户同名的目录,作为用户的宿主目录)

INACTIVE=-1 //是否启用户过期停止使用权,-1代表不启用

EXPIRE= //设定过期时间,格式为20170501

SHELL=/bin/bash //设定用户的默认使用shell

SKEL=/etc/skel //新用户宿主目录模板目录

CREATE_MAIL_SPOOL=yes //是否为用户启用邮件通知功能

创建新用户时,可以设置新用户可登陆系统的时间范围(2017-5-1)

/etc/skel:目录(用户宿主目录模板目录,只对新建用户生效)

.bash_logout:用户注销时执行的命令

.bash_profile:用户登录系统时执行的命令(用户变量)

.bashrc:用户登录一个新shell时执行的命令

/etc/login.defs文件解释:(只对新建用户生效)

MAIL_DIR /var/spool/mail //用户系统邮件存放目录
PASS_MAX_DAYS 99999 //密码最长使用期限
PASS_MIN_DAYS 0 //最短使用期限,0代表不受限制
PASS_MIN_LEN 5 //密码最小长度
PASS_WARN_AGE 7 //密码过期警告时间
UID_MIN 1000 //最小UID号
UID_MAX 60000 //最大UID号
SYS_UID_MIN 201 //系统用户最小UID
SYS_UID_MAX 999 //系统用户最大UID
GID_MIN 1000 //普通组最小GID
GID_MAX 60000 //普通组最大GID
SYS_GID_MIN 201 //系统组最小GID
SYS_GID_MAX 999 //系统组最大GID
CREATE_HOME yes //是否创建宿主目录
UMASK 077 //关于权限反掩码
USERGROUPS_ENAB yes //删除用户时是否删除组
ENCRYPT_METHOD SHA512 //用户密码的加密方式

用户管理:
useradd, userdel, usermod, passwd, chsh, chfn, finger, id, chage

# yum -y install vim make cmake man man-pages mlocate sysstat net-tools
# yum -y groupinstall "Development tools"
# mandb //生成whatis数据库
# updatedb //生成locate数据库
# vim /etc/selinux/config
SELINUX=disabled

useradd:添加新用户(更改/etc/passwd /etc/shadow /etc/group)
useradd [options] user_name
-c:为用户添加描述信息

-d:指定用户的宿主目录(默认宿主目录在/home目录)

-D:查看和改变默认的值(修改/etc/default/useradd文件中的选项和值)

-g:修改组

-b:修改宿主目录

-f:修改过期是否停用

-e:修改过期 时间

-s:修改默认shell

-e:新建用户时,设置用户默认的过期时间,时间格式为:YYYY-MM-DD -g:指定用户的基本组(没有该选项,Linux新建用户时,会新建一个同名组作为用户的基本组)

-G:新建用户时指定用户的附加组,附加组可以有多个,用逗号隔开

-m:创建宿主目录,和-k一起使用

-M:创建用户时,不为用户创建宿主目录

-p:创建用户时,为用户设置加密的密码(不推荐使用)

-r:创建系统用户(不创建宿主目录,UID和GID使用系统用户ID)

-s:指定默认的shell(系统支持的shell)

-u:指定用户的UID

passwd:为用户设置密码/锁定解锁用户/查看状态

语法:passwd [options] user_name

root用户可以为普通用户设置密码

-l:锁定用户(暂时无法登陆系统)

-u:解锁用户

-S(大写):查看用户状态,显示/etc/shadow文件中各个字段的内容

--stdin:标准输入,经常用在shell脚本编程中为用户自动设置密码

-d:删除用户密码,允许普通用户以空密码登录(不建议使用)

-e:设置用户密码过期,用户再次登录时,需要修改密码

-n:设置密码最小使用期限,修改/etc/shadow中第四列

-x:设置密码最大使用期限,修改/etc/shadow中第五列

-w:设置密码过期前的警告时间,修改/etc/shadow第六列

-i:修改过期后的宽限时间,修改/etc/shadow第七列

普通用户设置密码时,先验证当前密码,符合密码策略

root用户设置密码时,不需要验证当前密码,并且也可以不要求符合密码策略

/etc/shadow:保存用户密码相关信息的文件:

第一列:用户登录名
第二列:加密密码(idsalt$encrypted password)
第三列:最后一次更改密码时间
第四列:密码最小使用时间(用户无法更改密码)
第五列:密码最大使用期限(用户必须更改密码的时间)
第六列:密码过期前的警告时间
第七列:密码过期后一个宽限时间
第八列:密码失效时间
第九列:保留

userdel:删除用户(修改/etc/passwd /etc/shadow /etc/group文件)
语法:userdel [options] user_name
-r:删除用户时,连同删除用户的宿主目录

usermod:修改用户属性(修改/etc/passwd中内容)
语法:usermod [options] user_name
-c:更改用户的描述信息
-d:更改用户的宿主目录
-e:更改用户的过期时间,格式YYYY-MM-DD
-f:修改/etc/shadow第七列内容
-g:修改用户的基本组
-G:修改用户的附加组
-l:修改用户的登录名
-s:修改用户的shell(系统支持的shell程序)
-L(大写):锁定用户
-U(大写):解锁用户

chsh:修改用户的shell
语法:chsh [options] user_name
-s:后面跟shell,更改用户的shell
-l:显示当前系统支持的shell(/etc/shells)

finger:显示用户的基本信息
finger [options] user_name

chfn:修改用户的基本信息
  语法:chfn [options] user_name
  -o:修改办公室
  -p:修改办公室电话号码
  -h:修改家庭电话号码
  -f:修改用户名

id:显示用户和组的ID
  语法:id [options] user_name
  -a:忽略其他版本区别
  -Z(大写):显示安全上下文内容(selinux)
  -g:显示有效组的ID
  -G:显示所有组的ID
  -n:不显示组号,显示名字
  -u:显示用户ID

chage:修改用户密码的时间信息(/etc/shadow)
  语法:chage [options] user_name
  -l:列出用户的详细的密码参数
  -d:修改/etc/shadow第三列内容,后面跟日期,格式YYYY-MM-DD
  -E:修改/etc/shadow第八列内容,后面跟日期,格式YYYY-MM-DD
  -I:修改/etc/shadow第七列内容,后面跟天数
  -m:修改/etc/shadow第四列内容,后面跟天数
  -M:修改/etc/shadow第五列内容,后面跟天数
  -W:修改/etc/shadow第六列内容,后面跟天数

组管理:
    groupadd, groupdel, groupmod, gpasswd,newgrp,groupmems

组分类:
管理员组:root
普通用户组:
系统用户组:

按照用户分类:

管理员组:root
基本组:初始化的组
附加组:有效组

组:逻辑容器,存放用户,实现集中授权
配置文件:
/etc/group:
第一列:组的名字
第二列:组的密码区域
第三列:组的ID号(GID)
第四列:组中的成员
/etc/gshadow:
第一列:组的名字
第二列:组的密码
第三列:组的管理员
第四列:组成员列表

groupadd:
-d:创建组时指定组的ID
-r:添加一个系统组

groupdel:删除一个组
默认情况下,不能删除一个用户的基本组,可以通过修改用户的基本组后删除该组
可以删除用户的同时,系统会删除同名基本组(该组是一个用户的基本组)

groupmod:修改组的属性
-g:修改组的GID号
-n:为组重命名

gpasswd:为组设置密码,设置组的管理员,添加组成员
-A:为组设置管理员
-M:把一些用户添加组中,用逗号隔开(普通用户)
-r:为组移除密码
-R:让组的密码失效
-a:为组添加成员(组的管理员操作命令)

-d:删除组成员(组的管理员操作命令)

newgrp:为用户修改有效组(只在当前shell生效)
exit:退出newgrp

groupmems:为组添加成员,显示组成员列表

权限:是用户对于系统资源及数据等操作的能力

三类用户权限:
属主:owner
属组:group
其他:other

-rw-r--r--:10位
第一位:文件类型
第2-4位:文件属主权限
第5-7位:文件属组权限
第8-10位:其他用户的权限

文件:
r(read):读权限,可以查看文件的内容,使用cat more等命令
w(write):写权限,可以修改文件的内容及删除文件
x(execute):执行权限,可执行文件或脚本

目录:
r:读权限,可以查看目录中的列表,执行ls命令
w:写权限,可以向目录中新建文件或目录等
x:执行权限,可以执行cd命令

权限用数字表示:
r:4
w:2
x:1
---:000 0

rwx:111

crw-rw-rw- 1 root root   1,   9 Jun 11 16:17 urandom
666

drwxr-xr-x 2 root root   0 Jun 11 16:17 pts
755

crw-rw-rw- 1 root tty   5,   0 Jun 11 16:17 tty
666

lrwxrwxrwx 1 root root   15 Jun 11 16:17 stderr -> /proc/self/fd/2
777

chown:change owner改变文件属主和属组
-R:递归更改目录的属主或属组
--reference:复制一个目录的权限属性到目标文件
chown:修改文件属组时,在组名前添加点符号(.)或冒号(:)

同时修改属主和属组:chown username.groupname filename

chgrp:change group修改文件属组

chmod:修改文件或目录权限
语法:
chmod [augo][+-=][rwx] file_name
-R:递归更改文件或目录的权限

a:所有用户
u:文件属主
g:文件属组
o:其他用户

+:为用户添加权限

-:为用户减少权限
=:为用户赋予权限

-rw-r--r-- 1 root root 0 Jun 11 17:05 b.txt
root:rwx =111=7
root:组 rw-=110=6
other:rw-=110=6

766

默认情况下,为了安全,Linux系统新建文件时不添加执行权限

umask:反掩码(在/etc/login.defs中定义)

rwx:7
文件:rw-rw-rw-:666 644=rw-r--r--
目录:rwxrwxrwx:777 755=rwxr-xr-x
新建文件时,权限掩码(666)减去反掩码(022)就是新建文件的权限(644)
新建目录时,权限掩码(777)减去反掩码(022)就是新建目录的权限(755)

Linux隐藏权限:

chattr:修改文件的隐藏权限,/ /dev/ /tmp /var/ /etc/目录不受chattr保护

语法:
chattr +-= [options] file_name

a:只能向文件中追加数据,不能删除
i:任何用户不能删除文件
A:不同步访问时间
+:添加相应权限
-:删除相应权限
=:设定相应权限

lsattr:查看文件或目录的隐藏权限

/etc/passwd /etc/shadow /etc/group

tom:/etc/passwd /etc/shadow /etc/group添加信息,新建宿主目录(修改宿主目录的归属)

echo "tom:x:1001:1001::/home/tom:/bin/bash" >> /etc/passwd

认证机制:
3A认证:
    认证机制:authentication(资源申请者证明自己本身的一个过程)
        授权机制:authorization(用户是否可以访问一个服务或文件,用户访问服务或文件的能力)
        审计机制:audition

root a.txt rw-
login:user_name:passwd

user_name--UID /etc/passwd
www.baidu.com---IP /etc/hosts dns
httpd---80

名称解析:字符解析成数字 数字解析成字符
        file
        dns
        db(数据)
        ldap(轻量级目录服务访问协议)
        nis:网络信息服务,开源软件
        nisplus:sun公司商业软件

设计内核时,添加相应代码完成解析

应用程序 nsswitch(name services switch) 名称解析库(.so)

nsswitch配置文件:/etc/nsswitch.conf

info:method[[action]] mehtod[[action]]

passwd
goroup
shadow

readelf:查看ELF文件
 -a:查看所有

一、什么是nsswithch.conf(服务搜索顺序)文件呢nsswitch.conf(name service switch configuration,名字服务切换配置)文件位于/etc目录下,由它规定通过哪些途径以及按照什么顺序以及通过这些途径来查找特定类型的信息,还可以指定某个方法奏效或失效时系统将采取什么动作。
Nsswitch.conf中的每一行配置都指明了如何搜索信息,每行配置的格式如下:
Info: method[[action]] [method[[action]]...]
其中,info指定该行所描述的信息的类型,method为用来查找该信息的方法,action是对前面的method返回状态的响应。action要放在方括号里。
二、nsswitch.conf的工作原理
 当需要提供nsswitch.conf文件所描述的信息的时候,系统将检查含有适当info字段的配置行。它按照从左向右的顺序开始执行配置行中指定的方法。在默认情况下,如果找到期望的信息,系统将停止搜索。如果没有指定action,那么当某个方法未能返回结果时,系统就会尝试下一个动作。有可能搜索结束都没有找到想要的信息。
1、信息(Info)
 Nsswitch.conf文件通常控制着用户(在passwd中)、口令(在shadow中)、主机IP和组信息(在group中)的搜索。下面的列表描述了nsswitch.conf文件控制搜索的大多数信息(Info项)的类型。
automount:自动挂载(/etc/auto.master和/etc/auto.misc)
bootparams:无盘引导选项和其他引导选项(参见bootparam的手册页)
ethers:MAC地址
group:用户所在组(/etc/group),getgrent()函数使用该文件
hosts:主机名和主机号(/etc/hosts),gethostbyname()以及类似的函数使用该文件

networks:网络名及网络号(/etc/networks),getnetent()函数使用该文件
passwd:用户口令(/etc/passwd),getpwent()函数使用该文件
protocols:网络协议(/etc/protocols),getprotoent()函数使用该文件
publickey:NIS+及NFS所使用的secure_rpc的公开密钥
rpc:远程过程调用名及调用号(/etc/rpc),getrpcbyname()及类似函数使用该文件
services:网络服务(/etc/services),getservent()函数使用该文件
shadow:映射口令信息(/etc/shadow),getspnam()函数使用该文件
aiases:邮件别名,sendmail()函数使用该文件

2、方法(method):下面列出了nsswich.conf文件控制搜索信息类型的方法,对于每一种信息类型,都可以指定下面的一种或多种方法:
files:搜索本地文件,如/etc/passwd和/etc/hosts
nis:搜索NIS数据库,nis还有一个别名,即yp
dns:查询DNS(只查询主机)
compat:passwd、group和shadow文件中的±语法

3、搜索顺序(从左至右)两个或者更多方法所提供的信息可能会重叠。举例来说,files和nis可能都提供同一个用户的口令信息。如果出现信息重叠现象,就需要考虑将哪一种方法作为权威方法(优先考虑),并将该方法放在方法列表中靠左的位置上。
默认nsswitch.conf文件列出的方法并没有动作项,并假设没有信息重叠(正常情况)。在这种情况下,搜索顺序无关紧要:当一种方法失败之后,系统就会尝试下一种方法,只是时间上受到一点损失。如果在方法之间设置了动作,或者重叠的项的内容不同,那么搜索顺序就变得重要起来。
例如下面两行nsswitch.conf文件配置行:
passwd files nis
host nis files dns
第一行让系统在/etc/passwd文件中搜索口令信息,如果失败的话,就使用NIS来查找信息。如果正在查找的用户同时出现在这两个地方,就会使用本地文件中的信息,因此它就是权威信息。第二行先使用NIS搜索；如果失败的话,就搜索/etc/hosts文件；如果再次失败的话,核对DNS以找出主机信息。
4、动作项([action])
在每个方法后面都可以选择跟一个动作项,用来指定如果由于某种原因该方法成功抑或失败需要做些什么。动作项的格式如下:
[[!]STATUS =action]

其中,开头和末尾的方括号属于格式的一部分,并不是用来指出括号中的内容是可选的。
STATUS(按照约定使用大写字母,但本身并不区分大小写)是待测试的状态,action是如果STATUS匹配前面的方法所返回的状态将要执行的动作。开头的感叹号(!)是可选的,其作用是将状态取反。
STATUS的取值如下:
NOTFOUND:方法已经执行,但是并没有找到待搜索的值。 默认的动作是continue。
SUCCESS:方法已经执行,并且已经找到待搜索的值,没有返回错误。默认动作是return。
UNAVAIL:方法失败,原因是永久不可用。举例来说,所需的文件不可访问或者所需的服务器可能停机。默认的动作是continue。
TRYAGAIN:方法失败,原因是临时不可用。举例来说,某个文件被锁定,或者某台服务器超载。默认动作是continue。
action的取值如下:
return:返回到调用例程,带有返回值,或者不带返回值。
continue:继续执行下一个方法。任何返回值都会被下一个方法找到的值覆盖。
示例:
举例来说,下面这行取自nsswitch.conf文件,它的作用是让系统首先使用DNS来搜索给定主机的IP地址。DNS方法后面的动作项是测试该方法所返回的状态是否为“非(!)UNAVAIL”。
hosts dns [!UNAVAIL=return] files
如果DNS方法没有返回UNAVAIL(!UNAVAIL),也就是说DNS返回SUCCESS、NOTFOUND或者TRYAGAIN,那么系统就会执行与该STATUS相关的动作(return)。其结果就是,只有在DNS服务器不可用的情况下才会使用后面的方法(files)。
如果DNS服务器并不是不可用(两次否定之后就是“可用”),那么搜索返回域名或者报告未找到域名。只有当服务器不可用的时候,搜索才会使用files方法(检查本地的/etc/hosts文件)。

5、compat方法:passwd、group和shadow文件中的"±"

可以在/etc/passwd、/etc/group和/etc/shadow文件中放入一些特殊的代码,(如果在nsswitch.conf文件中指定compat方法的话)让系统将本地文件和NIS映射表中的项进行合并和修改。

在这些文件中,如果在行首出现加号' + ',就表示添加NIS信息;如果出现减号' - ',就表示删除信息。举例来说,要想使用passwd文件中的这些代码,可以在nsswitch.conf文件中指定passwd: compat。然后系统就会按照顺序搜寻passwd文件,当它遇到以+或者 开头的行时,就会添加或者删除适当的NIS项。

虽然可以在passwd文件的末尾放置加号,在nsswitch.conf文件中指定passwd: compat,以搜索本地的passwd文件,然后再搜寻NIS映射表,但是更高效的一种方法是在nsswitch.conf文件中添加passwd: file nis而不修改passwd文件。

nsswitch.conf:名称解析

login:
root--nsswitch.conf--passwd: files
123--nsswitch.conf--shadow: files
authentication:123--sha512(salt)--/etc/shadow compare

名称解析:libnss库
authentication:独立验证机制

名称解析和验证是两套独立运行的机制

nsswitch:files db nis ldap
authentication:files db nis ldap kerberos等

authentication:

PAM:嵌入式认证模块,PAM本身不会执行验证,借助策略和模块来验证

ldd:查看某个服务所支持的模块
    语法:ldd service_path

/etc/pam.d/service_name(系统中支持pam认证的服务名称)
/etc/pam.d/other(当系统中支持pam认证的服务没有匹配到任何条目时,匹配other中的配置)

/etc/pam.d/service_name格式

type control modules [modules_args]

type:
auth(entication):身份验证,匹配用户名和密码
account:检查用户名和密码的有效性
password:检查修改改密码时,密码是否符合标准

session:检查用户会话相关属性

control:
required:一票否决权,继续匹配下面的条目,最后不匹配(保护系统的安全性)
requisite:一票否决权,不匹配下面的条目,直接给用户返回结果
sufficient:一票同意权,不匹配下面的条目,直接给用户返回结果(慎用)
optional:不影响最终结果
include:引用其他的配置文件,把匹配权交给其他配置文件(如果其他配置跳出,则整个匹配结束)
substack:引用其他的配置,把匹配权交给其他配置文件(如果其他配置跳出,则仅仅跳出子匹配)

modules:匹配认证模块

另外还有一种比较复杂的格式为value = action的语法来设置控制标志,标志之间会以空格分开.格式如下:
value1 = action1 value2 = action2 ……
其中value可以是下列Linux PAM库的返回值:
success、open_err、symbol_err、service_err、 system_err、buf_err、
perm_denied、auth_err、cred_insufficient、authinfo_unavail、user_unknown、
maxtries、new_authtok_reqd、acct_expired、 session_err、cred_unavail、
cred_expired、cred_err、no_module_data、conv_err、 authtok_err、
authtok_recover_err、authtok_lock_busy、authtok_disable_aging、 try_again、
ignore、abort、authtok_expired、module_unknown、bad_item和default.
最后一个(default)能够用来设置上面的返回值无法表达的行为.

required [success=ok new_authtok_reqd=ok ignore=ignore default=bad]
requisite [success=ok new_authtok_reqd=ok ignore=ignore default=die]
sufficient [success=done new_authtok_reqd=done default=ignore]
optional [success=ok new_authtok_reqd=ok default=ignore]

action值:
ignore:忽略执行结果

bad:如果失败,结果被用于整个执行栈,后续栈继续执行
die:和bad相似,但是失败直接返回结果,后续栈不执行
ok:如果PAM_SUCCESS覆盖之前值
done:和ok 相似,但是栈直接返回结果
reset:重置栈当前状态

modules:PAM认证模块:
/etc/pam.d/*:每一个应用程序PAM配置文件
/usr/lib64/security/*:PAM认证模块文件的实际存放位置
/etc/security/*:其他PAM环境的配置文件,比如 access.conf
/usr/share/doc/pam-x:存放详细的PAM的说明文件

查看应用程序是否支持PAM认证:
ldd app_path
如: ldd /usr/sbin/vsftpd | grep libpam

pam_env.so:设置环境变量的模块,如果需要额外的环境变量,可以使用pam_env.so进行设置,配置文件/etc/security/pam_env.conf

pam_unix.so:提供验证阶段的验证功能,也可以提供授权管理

pam_secruetty.so:限制系统管理员只能够从安全的终端登录,安全终端:/etc/securetty

pam_nologin.so:限制普通使用者能否登录主机进行使用,当/etc/nologin文件存在时,所有的普通用户都无法登录

pam_cracklib.so:可是限制恶意攻击,检查密码强度

pam_pwquality.so:完全兼容pam_cracklib.so,检查密码强度,包括设置的密码是否在字典

pam_limits.so:限制使用者打开文件数量,单个文件大小等

pam_rootok.so:如果UID为0,直接通过,如root用户su到普通用户使用此模块

pam_listfile.so:验证使用其他的文件

pam_access.so:控制访问的模块,默认配置/etc/security/access.conf
  - : ALL EXCEPT gooann:ALL

ulimit:语法
 # ulimit [options] 限额
options:
 -H:hard limit:严格的限制,必须不能超过这个设置的数值
 -S:soft limit:警告的限制,超过这个数值系统会发出警告,通常soft比hard小
 -a:后面不加任何选项,可以列出系统中所有的限制额度
 -c:当某个用户程序发生错误的时候,系统会将内存中的应用程序写成文件,这种文件成为核心文件(core file),限制每个核心文件的最大容量
 -f:此shell可以创建的文件的大小,单位kb
 -l:可以锁定的内存量
 -u:单一用户可以使用的最大的进程量
 -t:可以使用的最大的CPU时间

limits配置文件:/etc/security/limits.conf
配置文件格式
<domain> <type> <item> <value>

domain:可以是用户,组或*,*代表所有限制
type:soft(软限制) hard(硬限制)
item:
- core - limits the core file size (KB)
- data - max data size (KB)
- fsize - maximum filesize (KB)
- memlock - max locked-in-memory address space (KB)
- nofile - max number of open file descriptors
- rss - max resident set size (KB)
- stack - max stack size (KB)
- cpu - max CPU time (MIN)
- nproc - max number of processes
- as - address space limit (KB)

- maxlogins - max number of logins for this user
- maxsyslogins - max number of logins on the system
- priority - the priority to run user process with
- locks - max number of file locks the user can hold
- sigpending - max number of pending signals
- msgqueue - max memory used by POSIX message queues (bytes)
- nice - max nice priority allowed to raise to values: [-20, 19]

value:限制的具体数值

试验:使用pam认证模块限制sshd登录用户(pam_listfile.so)

只允许root和gooann组的用户可以登录

locate find tar gzip bzip tree stat install du df uname less more date clock hwclock