

Linux 云计算集群架构师

学神 IT 教育：从零基础到实战，从入门到精通！

版权声明：

本系列文档为《学神 IT 教育》内部使用教材和教案，只允许 VIP 学员个人使用，禁止私自传播。否则将取消其 VIP 资格，追究其法律责任，请知晓！

免责声明：

本课程设计目的只用于教学，切勿使用课程中的技术进行违法活动，学员利用课程中的技术进行违法活动，造成的后果与讲师本人及讲师所属机构无关。倡导维护网络安全人人有责，共同维护网络文明和谐。

联系方式：

学神 IT 教育官方网站: <http://www.xuegod.cn>

Linux 云计算架构师进阶学习群 QQ 群: 1072932914



学习顾问：小语老师

学习顾问：边边老师

学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新动态，获取更多学习资料及答疑就业服务！

第六章 Centos8 用户管理

本节所讲内容:

- 6.1 用户和组的相关配置文件
- 6.2 管理用户和组
- 6.3 实战: 进入 centos8 紧急模式恢复 root 密码

用户一般来说是指系统的使用者, 使用者可以使用这些名称来登录使用计算机, 除了使用者之外, 一些系统服务也需要含有部分特权的用户账户运行; 因此出于安全考虑, 用户管理应运而生, 它加以明确限制各个用户账户的权限, **root 在计算机中用拥有至高特权**, 所以一般只作管理用, 非特权用户可以通过 SU 或 SUDO 程序来临时获得特权。

Linux 系统通过用户和用户组实现访问控制, 包括对文件访问、设备使用的控制。

1 人可以拥有很多账户, 只是彼此名称不同, 比如 root 名称已经占用就不能再用了, 此外, 任意用户可能从属某个用户组, 此用户可以加入某些已经存在的组来获得该组的特权。

每个文件的属性中都有一个文件拥有者和所属组。另外, 还有三种类型的访问权限: 读 (read)、写 (write)、运行 (execute)。我们可以针对文件的属主、属组、而设置相应的访问权限。再次, 我们可以通过 ll 或 stat 命令查询文件属主、属组和权限。

```
[root@xuegod63 ~]# ll | tail -2
-rw-----. 1 root root 1374 3 月 12 17:34 anaconda-ks.cfg
-rw-r--r--. 1 root root 1529 3 月 12 18:10 initial-setup-ks.cfg
[root@xuegod63 ~]# stat anaconda-ks.cfg
文件: "anaconda-ks.cfg"
大小: 1680      块: 8          IO 块: 4096   普通文件
设备: 803h/2051d Inode: 16797763 硬链接: 1
权限: (0600/-rw-----)  Uid: ( 0/   root)  Gid: ( 0/   root)
```

6.1 用户账号

6.1.1 用户的分类

Linux 用户三种角色: 超级用户, 普通用户, 虚拟用户。

超级用户: root 拥有对系统的最高的管理权限, UID=0

普通用户:

系统用户 UID 范围: 1-999 (centos7/8 版本) 1-499 (centos6 版本)

本地用户 UID 范围: 1000+ (centos7/8 版本) 500+ (centos6 版本)

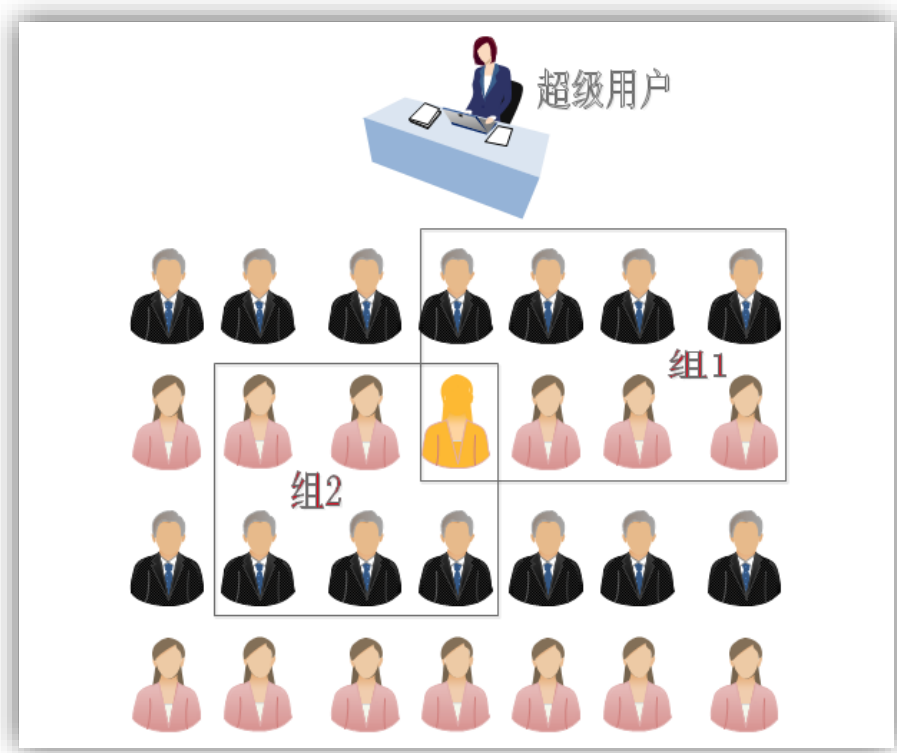
UID: 即每个用户的身份标示, 类似于每个人的身份证号码。

虚拟用户: 伪用户 一般不用来登录系统的, 它主要是用于维持某个服务的正常运行。如: ftp, apache

用户基本信息保存位置: /etc/passwd

下图是用户和组的关系:

- 一对一: 一个用户可以存在一个组中; 一对多: 一个用户可以存在多个组中
- 多对一: 多个用户可以存在一个组中; 多对多: 多个用户可以存在多个组中



6.1.2 配置文件

名 称	文件路径	说 明
用户配置文件	/etc/passwd	记录了每个用户的一些基本属性，并且对所有用户可读，每一行记录对应一个用户
用户组文件	/etc/group	用户组信息存放文件，并且组名不能重复
用户密码信息文件	/etc/shadow	因为 passwd 文件对所有用户是可读的，为安全起见把密码从 passwd 中分离出来放入这个单独的文件，该文件只有 root 用户拥有读权限，从而保证密码安全性

6.2 用户管理

6.2.1 用户命令

命令: useradd

useradd -d -u "UID" -g "初始组" -G "附加组" -s "登陆的 shell" 用户

-u: 指定 uid

-d: -d 用户主目录路径，可以指定用户家目录

-M: 不创建用户的家目录

-g: 设置用户初始组的名称或数字 ID；该组必须是存在的；如果没有设置该选项，useradd 会根据 /etc/login.defs 文件中的 USERGROUPS_ENAB 环境变量进行设置。默认 USERGROUPS_ENAB yes 会用和用户名相同的名字创建群组。

-G: 用户要加入的附加组列表；使用逗号分隔多个组，不要添加空格；如果不设置，用户仅仅加入初始组。（一个用户只允许有一个主组，可以有多个附属组）

-s: 用户默认登录 shell 的路径；启动过程结束后，默认启动的登录 shell 在此处设定；请确保使用的 shell 已经安装，默认是 Bash。有时候需要禁止某些用户执行登录动作，例如用来执行系统服务的用

户。将 shell 设置成 /sbin/nologin 就可以禁止用户登录。

6.2.2 添加登录用户

例：添加一个名为 harry 的用户，并使用 bash 作为登录的 shell

```
[root@xuegod63 ~]# useradd harry
```

```
[root@xuegod63 ~]# tail -1 /etc/passwd
```

```
harry:x:1001:1001::/home/harry:/bin/bash
```

说明：此命令会自动创建 harry 组，并成为 harry 用户的默认主组，同时默认的登录 shell 是 bash
用户帐户的全部信息被保存在 /etc/passwd 文件。这个文件以如下格式保存了每一个系统帐户的所有信息（字段以 “:” 分割）

/etc/passwd: 每个字段的作用:

例如: root:x:0:0:root:/root:/bin/bash

用户名 : 密码占位符 : UID : GID : 用户描述 : 用户主目录 (bash 中 "~" 代表哪个) : 登录后使用的 shell

harry:x:1001:1001::/home/harry:/bin/bash #每行含意如下:

harry: 用户名

x: 密码占位符

1001: 用户的 UID, 使用正整数表示, 范围可以是 0-65535

1001: 用户所属组的 GID, 它都是用数字来表示的

用户描述信息: 对用户的功能或其它来进行一个简要的描述, 此字段会出现在登录用户界面, 可以通过点击输入对应密码进行登录, 却并不能手动输入此字段代替用户名。

/home/harry: 用户主目录 (shell 命令提示符中用 “~” 表示)

/bin/bash: 用户登录系统后使用的 shell

例：查看系统中, 支持哪些 shell

```
[root@xuegod63 ~]# cat /etc/shells #查看系统中, 支持哪些 shell
```

```
/bin/sh
```

```
/bin/bash
```

```
/usr/bin/sh
```

```
/usr/bin/bash
```

再安装一种名叫 zsh 的 shell

```
[root@xuegod63 ~]# cat /etc/shells
```

```
/bin/sh
```

```
/bin/bash
```

```
/usr/bin/sh
```

```
/usr/bin/bash
```

```
/usr/bin/zsh
```

```
/bin/zsh
```

```
[root@xuegod63 ~]# yum install zsh -y
```

```
[root@xuegod63 ~]# zsh
```

```
[root@xuegod63]~# cd /etc/sysconfig/network-scripts
```

```
[root@xuegod63]/etc/sysconfig/network-scripts#
```

注: zsh 这种 shell 会显示绝对路径的

6.2.3 指定用户 UID :

```
useradd -u 用户 ID
[root@xuegod63 ~]# useradd -u 1100 oracle
[root@xuegod63 ~]# id oracle
uid=1100(oracle) gid=1100(oracle) 组=1100(oracle)
[root@xuegod63 ~]# tail -1 /etc/passwd
oracle:x:1100:1100::/home/oracle:/bin/bash
[root@xuegod63 ~]# ls -a /home/oracle/
. .. .bash_logout .bash_profile .bashrc .mozilla
```

6.2.4 指定用户主目录

```
[root@xuegod63 ~]# useradd -d /opt/ftp ftp1
[root@xuegod63 ~]# ls -a /opt/ftp
[root@xuegod63 ~]# tail -1 /etc/passwd
ftp1:x:1101:1101::/opt/ftp:/bin/bash
```

6.2.5 指定用户的主组

例:

```
[root@xuegod63 ~]# useradd xuegod
[root@xuegod63 ~]# id xuegod
uid=1103(xuegod) gid=1103(xuegod) 组=1103(xuegod)
[root@xuegod63 ~]# useradd -g xuegod xuegod2
[root@xuegod63 ~]# id xuegod2
uid=1104(xuegod2) gid=1103(xuegod) 组=1103(xuegod)
```

6.2.6 指定用户的附加组

我们也可以把这个附属组称为补充组, 用户可以有 0 个或多个附加组的成员
如果一个组有多个成员, 我们是可以在/etc/group 文件中最后一个字段看到的

```
[root@xuegod63 ~]# useradd -G xuegod,oracle,root xuegod3
[root@xuegod63 ~]# id xuegod3
uid=1105(xuegod3) gid=1105(xuegod3)
组=1105(xuegod3),0(root),1001(harry),1103(xuegod)
[root@xuegod63 ~]# vim /etc/group
```

```
harry: x: 1001: xuegod3
oracle: x: 1100:
aaaa: x: 1101:
mk1: x: 1102:
xuegod: x: 1103: xuegod3
xuegod3: x: 1105:
"/etc/group" 721 10060
```

6.2.7 创建用户的另外一个命令

```
[root@xuegod63 ~]# adduser xuegod4
[root@xuegod63 ~]# id xuegod4
uid=1106(xuegod4) gid=1106(xuegod4) 组=1106(xuegod4)
[root@xuegod63 ~]# which adduser
/usr/sbin/adduser
[root@xuegod63 ~]# ll /usr/sbin/adduser
lrwxrwxrwx. 1 root root 7 9 月 19 2017 /usr/sbin/adduser -> useradd
注: adduser 是 useradd 的软链接
```

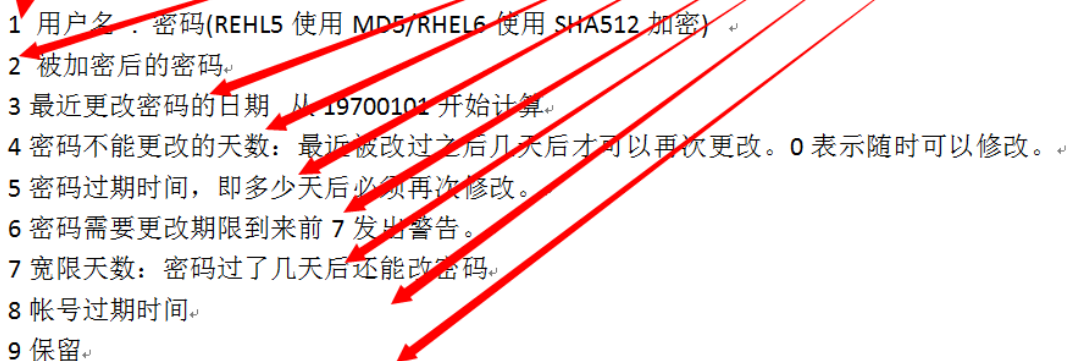
6.2.8 删除用户

语法: userdel [options] 用户名
选项: -r 删除的时候, 会同时删除用户的家目录和/var/mail 下的目录
[root@xuegod63 ~]# userdel -r xuegod3

6.2.9 密码的文件

```
[root@xuegod63 ~]# head -3 /etc/shadow
root:$6$C88LCVx5ZjfBU7xv$cKcdyNeTFmOYTs9NbRZDTA4hGcbMXc/5hQEWZKCtNyLqI
Bagrjct.pMfs39iEaF1UbEvcOzWZHMDf9Q5KojXM1::0:99999:7:::
```

```
root:$6$.BTyNB8Q397zR.KY$412.....M8ZHWiidd/:16274: 0 : 99999 : 7 : : :
```

- 
- 1 用户名 . 密码(REHL5 使用 MD5/RHEL6 使用 SHA512 加密)
 - 2 被加密后的密码
 - 3 最近更改密码的日期, 从 19700101 开始计算
 - 4 密码不能更改的天数: 最近被改过之后几天后才可以再次更改。0 表示随时可以修改。
 - 5 密码过期时间, 即多少天后必须再次修改。
 - 6 密码需要更改期限到来前 7 发出警告。
 - 7 宽限天数: 密码过了几天后还能改密码。
 - 8 帐号过期时间。
 - 9 保留。

格式如下:

name 登录名称, 这个必须是系统中的有效账户名

password 已加密密码, 分为三个部分, 第一部分是表示使用哪种哈希算法; 第二部分是用于加密哈希的 salt; 第三部分是已加密的哈希

哈希算法: \$1 表示 MD5 ; \$6 表示 SHA512 ; \$5 表示 SHA256

查看帮助说明: man 5 查看文件格式, man 3 查看库函数

man 5 passwd

man 5 shadow

man 5 group

lastchange 最近一次更改密码的日期, 以距离 1970/1/1 的天数表示

min-age 不能更改密码的最少天数, 最近更改过后几天才可以更改; 如果为 0 表示“最短期要求”

maxage 密码过期时间, 必须更改密码前的最多天数

warning 密码即将到期的警告期, 以天数表示, 0 表示 “不提供警告”

inactive 宽限天数, 密码到期后

expire 账号过期时间, 以距离 1970/1/1 的天数计算 (千年虫)

blank 预留字段

给用户添加密码:

```
[root@panda home]# passwd oracle
```

#交互式修改密码

Changing password for user oracle.

New password:

BAD PASSWORD: The password is shorter than 8 characters

Retype new password:

passwd: all authentication tokens updated successfully.

```
[root@xuegod63 ~]# echo 123456 | passwd --stdin xuegod
```

#不交互

```
[root@xuegod63 ~]# echo 123456 | passwd --stdin harry
```

互动: 两个用户的密码都是 123456 一样的, 那么 shadow 中加密的 hash 值一样吗?

答: 不一样。因为 salt (撒盐加密) 不一样

把 2 段加密的互换还能登陆吗? salt 什么时候指定的?

答: 可以登录, salt 在加密过程中随机生成, 保护了加密密码的机密性。

6.2.10 添加用户规则

控制添加用户规则文件的两个文件: /etc/default/useradd 和 /etc/login.defs

```
[root@panda home]# egrep -v "^\$|^#" /etc/login.defs
```

MAIL_DIR/var/spool/mail

PASS_MAX_DAYS 99999

#用户密码最长使用时间, 多少天后会有提醒

PASS_MIN_DAYS 0

#用户密码最短使用时间, 意思是多少天内不能修改密

码, 0 为不限制

PASS_MIN_LEN 5

#用户密码最小长度

PASS_WARN_AGE 7

#密码过期后会提醒多少天, 这些天内还没有修改密码的

用户, 账户会被冻结

UID_MIN 1000

#用户 ID 开始的数字

UID_MAX 60000

#用户 ID 结束的数字

SYS_UID_MIN 201

SYS_UID_MAX 999

GID_MIN 1000

GID_MAX 60000

#组 ID 结束的数字

SYS_GID_MIN 201

SYS_GID_MAX 999

CREATE_HOME yes

#是否为用户建立 home 目录

UMASK 077

USERGROUPS_ENAB yes

ENCRYPT_METHOD SHA512

#shadow 文件的加密算法

```
[root@panda home]# cat /etc/default/useradd
```

/etc/default/useradd 文件中的内容如下:

GROUP=100

#新建用户时默认初始组的 GID 号 (公共组), 现在使用的

都是私有组机制 (根据创建用户名称创建组)

HOME=/home	# /home 表示用户家目录的位置
INACTIVE=-1	#是否启用帐号过期。passwd 文件中第 7 列, -1 表示不启用
EXPIRE=	#帐号终止日期 shadow 中第 8 字段, 你可以直接设定帐号在

哪个日期后就直接失效, 而不理会密码的问题。

SHELL=/bin/bash	#默认 shell 使用哪个
SKEL=/etc/skel	#模板目录
CREATE_MAIL_SPOOL=yes	#是否创建邮箱文件

命令: chage 了解一下, 后期记不住时, 可以直接 vim 修改 shadow 文件

-m: 密码可更改的最小天数。为 0 代表任何时候都可以更改密码

-M: 密码保持有效的最大天数

-W: 用户密码到期前, 提前收到警告信息的天数

-E: 帐号到期的日期。过了这天, 此帐号将不可用

-d: 上一次更改的日期, 为 0 表示强制在下次登录时更新密码

例: 修改用户 xuegod 密码信息: 让这个用户 xuegod 首次登录系统时必须更改其密码

```
[root@xuegod63 ~]# chage -d 0 xuegod
```

```
[root@xuegod63 ~]# ssh xuegod@192.168.1.63
```

...

Are you sure you want to continue connecting (yes/no)? **yes**

Warning: Permanently added '192.168.1.63' (ECDSA) to the list of known hosts.

mk@192.168.1.63's password: **123456**

You must change your password now and login again! #提示必须改密码

更改用户 xuegod 的密码。

互动: 两个用户的 UID 可以一样吗?

```
[root@xuegod63 ~]# vim /etc/passwd # 改 xuegod uid 为 0
```

```
xuegod:x:0:0:xuegod:/home/mk:/bin/bash
```

```
[root@xuegod63 ~]# su - xuegod
```

上一次登录: 二 9 月 19 22:03:16 CST 2017:0 上

```
[mk@xuegod63 ~]# id xuegod
```

```
uid=0(mk) gid=0(root) 组=0(root),10(wheel)
```

查看用户相关命令:

#id 用户和组的信息

#whoami #查看当前有效用户名

#who #显示目前登入系统的用户信息。

#w #w 命令用于显示已经登陆系统的用户列表

#users #用于显示当前登录系统的所有用户的用户列表

6.2.11 修改用户信息

语法: usermod 【参数】用户名

常用参数:

-u UID

-d 宿主目录

-g	起始组	#只能有一个
-G	附加组	#可以有多个
-s	登录 shell	
-L	锁定	

例 1: 修改 UID

```
[mk@xuegod63 ~]# id oracle
uid=1100(oracle) gid=1100(oracle) 组=1100(oracle)
[mk@xuegod63 ~]# usermod -u 1111 oracle
[mk@xuegod63 ~]# id oracle
uid=1111(oracle) gid=1100(oracle) 组=1100(oracle)
```

例 2: 修改 shell

```
[root@panda home]# usermod -s /sbin/nologin oracle
[root@panda home]# grep oracle /etc/passwd
oracle:x:1111:1100:::/home/oracle:/sbin/nologin
```

例 3: 更改用户主目录

```
[root@panda home]# usermod -m -d /opt/aaa xuegod
-m 选项会自动创建新目录并且移内容到新目录里面
```

例 4: 添加用户描述信息

```
[root@xuegod63 ~]# usermod -c "hello world" xuegod
[root@xuegod63 ~]# grep xuegod /etc/passwd
mk:x:1000:1000:hello world:/opt/aaa:/bin/bash
```

总结: 如果你记不住命令, 那么直接改 vim /etc/passwd 一样的。

6.2.12 解决模板文件被删之后显示不正常的问题

```
[root@xuegod63 ~]# useradd xuegod88 && echo 123456 | passwd --stdin xuegod88
[root@xuegod63 ~]# echo 123456 | passwd --stdin xuegod88
[root@xuegod63 ~]# rm -rf /home/xuegod88/.bash*
[root@xuegod63 ~]# su - xuegod88
[xuegod88@xuegod63 ~]$
[xuegod88@xuegod63 ~]$
-bash-4.2$ exit      #在 centos6 或 7 会出现这个不完整的 shell 提示符, 如何处理?
[root@xuegod63 ~]# cp /etc/skel/.bash* /home/xuegod88/
[root@xuegod63 ~]# chown xuegod88:xuegod88 /home/xuegod88/.bash*
[root@xuegod63 ~]# su - xuegod88
```

6.3 实战: 进入紧急模式恢复 root 密码

6.3.1 实战场景找回 centos8 系统中 root 密码

公司一台 centos8 系统, 忘记 root 密码了, 需要你快速把 root 密码修改为 xuegod63, 找回 root 身份。

实验环境: 开启一台 centos8 系统

```
CentOS Linux (4.18.0-80.el8.x86_64) 8 (Core)
CentOS Linux (0-rescue-7f34f9a189db49419e342b49851e3fe4) 8 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
```

#开机时按任意键，即可进入此界面，光标选择第一条，`e`表示进入编辑模式

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-80.el8.x86_64 root=/dev/mapper/cl-root ro crashke\
rnel=auto resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb \
quiet
initrd ($root)/initramfs-4.18.0-80.el8.x86_64.img $tuned_initrd
```

#在 linux 行尾添加 **rd.break**

#进入编辑模式后会看到这些信息，默认情况，都是以 ro 只读方式引导系统进入

在 centos8 下，写的位置如下：

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.18.0-193.el8.x86_64 root=UUID=33968e4b-4074-404f-8aec-\
48ecad99653b ro resume=UUID=5a448cda-83d9-458b-88fa-139e01a2e525 rhgb quiet rd\
.break
initrd ($root)/initramfs-4.18.0-193.el8.x86_64.img $tuned_initrd _

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

改完之后，按下 Ctrl+X 进入紧急模式

原理：打断系统正常启动，然后进一个 bash 环境，系统并没有真正的启动

```
[ 1.293892] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.294519] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.295969] sd 0:0:0:0: [sda] Assuming drive cache: write through

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/#
```

emergency [i'm3:dʒənsi] 紧急

查看系统根挂载情况:

```
switch_root:/# mount
rootfs on / type rootfs (rw)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=975248k,nr_inodes=243812,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/sys/fs/cgroup/systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
configs on /sys/kernel/config type configs (rw,relatime)
/dev/sda3 on /sysroot type xfs (ro,relatime,attr2,inode64,noquota)
switch_root:/#
```

发现是只读的。需要重新以 rw 方式挂载/sysroot。

mount -o remount,rw /sysroot #重新挂载, 使其拥有读写权限

```
switch_root:/# mount -o remount,rw /sysroot/
switch_root:/#
```

换根, 修改密码

chroot 命令用来在指定的根目录下运行指令

chroot, 即 change root directory (更改 root 目录)。在 linux 系统中, 系统默认的目录结构都是以/, 即是以根 (root) 开始的。而在使用 chroot 之后, 系统的目录结构将以指定的位置作为/位置

在经过 chroot 命令之后, 系统读取到的目录和文件将不在是旧系统根下的而是新根下 (即被指定的新的位置) 的目录结构和文件。

```
switch_root:/# chroot /sysroot/
sh-4.4#
```

输入: LANG=en_US.UTF-8 #修改语言环境为英文, 这样显示乱码

LANG=zh_CN.UTF-8

passwd #开始修改密码

```
sh-4.4#  
sh-4.4# LANG=en  
sh-4.4# passwd  
Changing password for user root.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
sh-4.4#
```

注: 如果系统的 selinux 开启着, 则需要执行命令: touch /.autorelabel 以更新系统信息, 否则重启之后密码修改不会生效, 先退出当前根, reboot 重启系统。我们已经关闭 selinux, 不需要创建 /.autorelabel 。

```
sh-4.2# touch /.autorelabel  
sh-4.2# exit  
exit  
switch_root:/# reboot
```

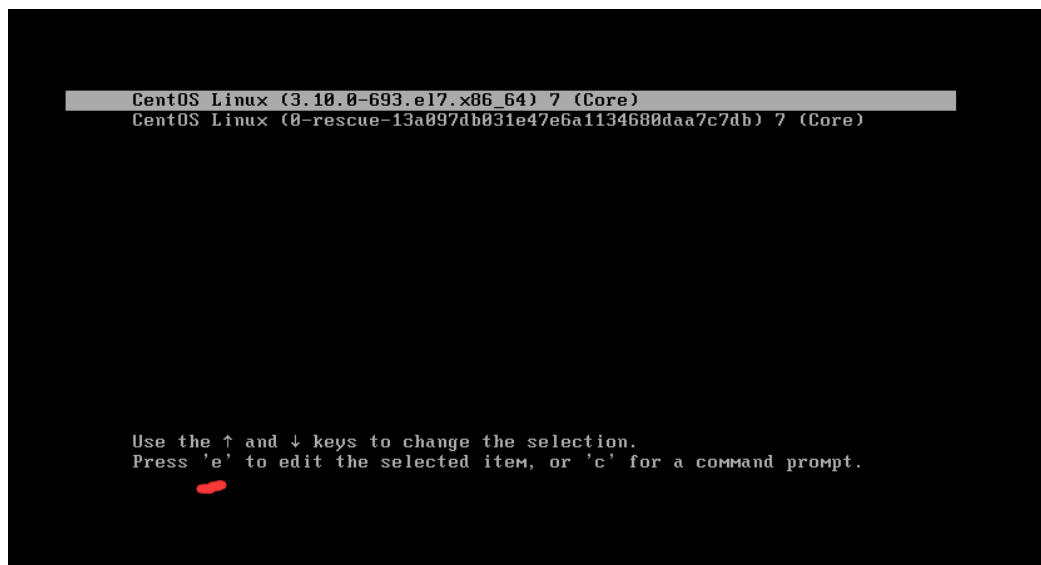
扩展: 创建此文件: 开启 selinux 的情况下需要创建此文件, 因为在 rd.break 环境下 SELinux 是不生效的。在不生效的情况下我们修改了用户的密码, 也就是修改了/etc/shadow 文件, 所以密码文件的安全上下文的特性会被取消。如果没有让系统在启动时自动恢复 SELinux 的安全上下文, 系统会报错“无法登录”, 所以 SELinux 在 Enforcing 模式下的时候 (如在 disabled 模式下则不用), 在根目录下 touch 隐藏文件 autorelabel 会让系统在重启时以 SELinux 默认类型重新写入 SELinux 安全上下文。

6.3.2 实战场景找回 centos7 系统中 root 密码

公司一台 centos7 系统, 忘记 root 密码了, 需要你快速把 root 密码修改为 xuegod, 找回 root 身份。

实验环境: 开启一台 Centos7 系统

首先重启, 按↑↓键, 进入如下界面, 选择第一项, 按下 e 键进行编辑



#进入编辑模式后会看到这些信息。找到“Linux16”开头的行, 在 Linux16 的行尾空格后添加“rd.break”

```
insmod part_msdos
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' a635d4d2-a\
21e-4d9b-b199-4c8d5cfed808
else
    search --no-floppy --fs-uuid --set=root a635d4d2-a21e-4d9b-b199-4c8d\
5cfed808
fi
linux16 /vmlinuz-3.10.0-693.2.2.el7.x86_64 root=UUID=4bcb433e-10e6-464\
d-a40b-00d018950149 ro rhgb quiet LANG=zh_CN.UTF-8 rd.break _
initrd16 /initramfs-3.10.0-693.2.2.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

```
insmod part_msdos
insmod xfs
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hin\
t-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 27ff55b7-6\
956-4ca8-a3b9-ee187cdf0528
else
    search --no-floppy --fs-uuid --set=root 27ff55b7-6956-4ca8-a3b9-ee18\
7cdf0528
fi
linux16 /vmlinuz-3.10.0-957.el7.x86_64 root=UUID=19d82030-4f25-416a-98\
a4-a6839d81adba rw rhgb quiet LANG=zh_CN.UTF-8 rd.break _
initrd16 /initramfs-3.10.0-957.el7.x86_64.img

Press Ctrl-x to start, Ctrl-c for a command prompt or Escape to
discard edits and return to the menu. Pressing Tab lists
possible completions.
```

这里改成 rw, 则不需要重新挂载, 直接以读写挂载

改完之后, 按下 Ctrl+X 进入紧急模式

原理: 打断系统正常启动, 然后进一个 bash 环境, 系统并没有真正的启动

```
[ 1.293892] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.294519] sd 0:0:0:0: [sda] Assuming drive cache: write through
[ 1.295969] sd 0:0:0:0: [sda] Assuming drive cache: write through

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/#
```

emergency [i'm3:d3ənsi] 紧急

查看系统根挂载情况:

```
switch_root:/# mount
rootfs on / type rootfs (rw)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,size=975248k,nr_inodes=243812,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/sys/fs/cgroup/systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/sda3 on /sysroot type xfs (ro,relatime,attr2,inode64,noquota)
switch_root:/#
```

发现是只读的。需要重新以 rw 方式挂载/sysroot。

mount -o remount,rw /sysroot #重新挂载,使其拥有读写权限

```
switch_root:/# mount -o remount,rw /sysroot/
switch_root:/#
```

换根, 修改密码

chroot 命令用来在指定的根目录下运行指令

chroot, 即 change root directory (更改 root 目录)。在 linux 系统中, 系统默认的目录结构都是以/, 即是以根 (root) 开始的。而在使用 chroot 之后, 系统的目录结构将以指定的位置作为/位置

在经过 chroot 命令之后, 系统读取到的目录和文件将不再是旧系统根下的而是新根下 (即被指定的新的位置) 的目录结构和文件。

```
switch_root:/# chroot /sysroot/
sh-4.4#
```

输入: LANG=en_US.UTF-8 #修改语言环境为英文, 这样不会显示乱码

passwd #开始修改密码

```
sh-4.4#
sh-4.4# LANG=en
sh-4.4# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
sh-4.4#
```

注: 如果系统的 selinux 开启着, 则需要执行命令: touch /.autorelabel 以更新系统信息让 SELinux 生效, 否则重启之后密码修改不会生效, 先退出当前根, reboot 重启系统。我们已经关闭 selinux, 不需要创建 /.autorelabel。


```
sh-4.2# touch /.autorelabel
sh-4.2# exit
exit
switch_root:/# reboot
```

扩展: 创建此文件: 开启 selinux 的情况下需要创建此文件, 因为在 rd.break 环境下 SELinux 是不生效的。在不生效的情况下我们修改了用户的密码, 也就是修改了/etc/shadow 文件, 所以密码文件的安全上下文的特性会被取消。如果没有让系统在启动时自动恢复 SELinux 的安全上下文, 系统会报错“无法登录”, 所以 SELinux 在 Enforcing 模式下的时候 (如在 disabled 模式下则不用), 在根目录下 touch 隐藏文件 autorelabel 会让系统在重启时以 SELinux 默认类型重新写入 SELinux 安全上下文。

扩展内容 (了解即可): SELinux

SELinux (Security Enhanced Linux 安全性增强的 Linux), 由美国国家安全局 NSA (National

Security Agency) 开发, 构建与 Kernel 之上, 拥有灵活的强制性访问控制结构, 主要用在提高 Linux 的安全性, 提供强健的安全保证, 可以防御未知攻击。

SELinux 是用于确定哪个进程可以访问哪些文件、目录和端口的一组安全规则。每个文件、进程、目录和端口都具有专门的安全标签, 称为 SELinux 上下文。

SELinux 标签具有多种上下文: User 用户、Role 角色、Type 类型和 Level 敏感度级别。目标策略会根据第三个上下文 (即 Type 类型上下文) 来制定自己的规则, 通常以 **t** 结尾

ls -Z

```
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

SELinux User	Role	Type	Level	File
unconfined_u	object_r	httpd_sys_content_t	s0	/var/www/html/file2

```
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 文档
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 下载
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 音乐
drwxr-xr-x. root root unconfined_u:object_r:admin_home_t:s0 桌面
[root@xuegod63 ~]#
```

↑ 用户 ↑ 角色 ↑ 类型 ↑ 级别

传统的 Linux 在没有 Selinux 保护的时候, 倘若运行于之上的服务器被骇客攻陷, 其服务器最高权限就可能随之丧失, 但是倘若有了 SELinux 的保护, 入侵的也只有服务本身, 整个服务器的最高权限依然还健在!

一句话熟悉 Selinux 的作用: 增强 Linux 系统安全性, 一个例子: 那么是我运行的 Apache 服务器被入侵, 也只是入侵到了我 httpd 这个服务, 可以把它禁锢到这里 (相当于整个服务器运行了 httpd, 入侵了 httpd 而已), 而系统的整个权限依然正常!

SELinux 模式

enforcing 强制模式, 只要 selinux 不允许, 就无法执行;

permissive 警告模式, 你可以执行, 但你所做事件全部记录;

disabled 关闭 selinux

```
[root@xuegod83 ~]# ls -Z /var/www/html
unconfined_u:object_r:httpd_sys_content_t:s0 a.txt
unconfined_u:object_r:httpd_sys_content_t:s0 index.html
unconfined_u:object_r:httpd_sys_content_t:s0 index.php
[root@xuegod83 ~]# ls -Z /home/yum
unconfined_u:object_r:user_home_t:s0 公共 unconfined_u:
unconfined_u:object_r:user_home_t:s0 模板 unconfined_u:
unconfined_u:object_r:user_home_t:s0 视频 unconfined_u:
unconfined_u:object_r:user_home_t:s0 图片 unconfined_u:
[root@xuegod83 ~]# ls -Z /root
unconfined_u:object_r:admin_home_t:s0 公共
unconfined_u:object_r:admin_home_t:s0 模板
unconfined_u:object_r:admin_home_t:s0 视频
unconfined_u:object_r:admin_home_t:s0 图片
unconfined_u:object_r:admin_home_t:s0 文档
```

selinux 控制的文件类型权限非常细, httpd 无权限访问其他文件类型, 黑客即使入侵了 httpd, 也没有权限访问其他类型文件, 不管你是不是属组属主有 rwx 权限, 都没用。

```
[root@xuegod83 ~]# mkdir /abc
[root@xuegod83 ~]# touch /abc/index.html
[root@xuegod83 ~]# ls -Z /abc/
unconfined_u:object_r:default_t:s0 index.html
[root@xuegod83 ~]# semanage fcontext -a -t httpd_sys_content_t '/abc(/.*)?'
semanage 管理
fcontext 声明文件的默认标签
-a 添加
-t 类型
/abc(/.*)'?是递归匹配/abc 目录下子文件和子目录
[root@xuegod83 ~]# ls -Z /abc/
unconfined_u:object_r:default_t:s0 index.html
[root@xuegod83 ~]# restorecon -RFvv /abc
restorecon 应用 fcontext 所声明的文件的标签
-R 递归
-F 强制
-vv 显示详细信息
[root@xuegod83 ~]# ls -Z /abc/
system_u:object_r:httpd_sys_content_t:s0 index.html
[root@xuegod83 ~]# semanage fcontext -d -t httpd_sys_content_t '/abc(/.*)?'
-d 删除
-t 类型

[root@xuegod83 ~]# restorecon -RFvv /abc
[root@xuegod83 ~]# ls -Z /abc/
system_u:object_r:default_t:s0 index.html
```

```
[root@xuegod83 ~]# semanage fcontext -l
```

-l 查看类型

实验 httpd 服务没有权限访问其他类型的文件

```
[root@xuegod63 ~]# yum -y install httpd
```

```
[root@xuegod63 ~]# systemctl start httpd
```

```
[root@xuegod63 ~]# echo hello >> /var/www/html/abc.txt
```

```
[root@xuegod63 ~]# ls -Z /var/www/html/abc.txt
```

```
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
```

/var/www/html/abc.txt

```
[root@xuegod83 ~]# curl 192.168.1.63/abc.txt
```

打开网页可以访问这个 abc.txt 的内容

```
[root@xuegod63 ~]# semanage fcontext -a -t default_t '/var/www/html/abc.txt'
```

```
[root@xuegod63 ~]# restorecon '/var/www/html/abc.txt'
```

```
[root@xuegod63 ~]# ls -Z '/var/www/html/abc.txt'
```

```
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 /var/www/html/abc.txt
```

```
[root@xuegod83 ~]# curl 192.168.1.63/abc.txt
```

打开网页没有权限访问这个 abc.txt 的内容

```
[root@xuegod63 ~]# semanage fcontext -d -t default_t '/var/www/html/abc.txt'
```

```
[root@xuegod63 ~]# restorecon '/var/www/html/abc.txt'
```

```
[root@xuegod63 ~]# ls -Z '/var/www/html/abc.txt'
```

```
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
```

/var/www/html/abc.txt

```
[root@xuegod83 ~]# curl 192.168.1.63/abc.txt
```

打开网页可以访问这个 abc.txt 的内容

总结:

- 6.1 用户和组的相关配置文件
- 6.2 管理用户和组
- 6.3 实战: 进入 centos8 紧急模式恢复 root 密码