

???????

???????

CentOS/RedHat)

```
groupadd -r asterisk
useradd -r -g asterisk -d /var/lib/asterisk -M asterisk
```

Ubuntu/Debian)

```
addgroup --system asterisk
adduser --system --ingroup asterisk --home /var/lib/asterisk --no-create-home --shell
/bin/bash asterisk
```

????

```
# RedHat
userdel -r <username>

# Debian
deluser <username> --remove-home
```

??????????

```
# Debian/Ubuntu
# Add the user into the group sudo
sudo usermod -aG sudo <user-name>
# Verify the user's groups
groups <user-name>
```

??????

????????????????????????????

```

# 锁定用户
usermod -L <username>

# 删除密码
# 删除密码
chage -d 0 <username>

# 解锁用户
usermod -U <username>

# 强制密码更新
chage -l <user-name>

```

??????

```

# 强制密码更新
chage -l <user-name>

# 强制密码更新
chage -M 10 <user-name>          # 10 天强制更新
chage -E "2017-02-20" <user-name> # 2017-02-20 到期
chage -I 10 <user-name>          # 每隔10天强制更新一次

# 强制密码更新
chage -E -1 <user-name>          ; 永久有效

```

???????

```

# 强制密码更新
usermod -L <user-name>
passwd -l <user-name>
chage -E 0 <user-name>

# 强制密码更新
usermod -U <user-name>
passwd -u <user-name>
chage -E <user-name>

# 强制密码更新
grep <user-name> /etc/shadow

```

```
dbtest:!$6$hFCW6eI1$kI9J9QrxCjnpvzFPJnxSpNvQ... 0000 ! 000000
```

```
# List the locked and passwordless accounts
```

```
getent shadow | awk '/^.*:[!\*].*/' | cut -d: -f1
```

```
“ ???Passwd ?????????????? SSH-Key ???
```

```
“ ?????????????????????? CronJob ???
```

???????????

RHEL 7

```
# To enable the faillock
# unlock_time: seconds
authconfig --enablefaillock --faillockargs="deny=5 unlock_time=1200" --update

# To disable the faillock
authconfig --disablefaillock --update

# Validate the configuration
authconfig --test

# Check the login attempts failed
faillock

# Unlock the user locked immediately
faillock --user test01 --reset
```

???????????

```
# 0000
usermod -c "John" john
# 00 shell
usermod -s "/sbin/nologin" alang
# 000000
```

```
usermod -l newuser currentuser
```

????????????

???????????????????? su ??????

????? devrpt ?????????????????????? su ? devrpt?

???: ?? sshd_config

```
# Added by Alang
# prevent certain users from using ssh for login
# while retaining the option to 'su username'
#
DenyUsers istdc
```

???: ??????????????????????

```
# ?? devrpt ???
passwd -d devrpt
```

???: ????????

? CentOS ???

1. ?? /etc/security/access.conf??????

```
# The line 'cron crond' is required
+:devrpt:cron crond tty1 tty2 tty3 tty4 tty5 tty6
-:devrpt:ALL
```

“ TIPS?

????? permission : username: origins

permission + ?? ? - ??

username ??

origins ??????? tty ??'??/????IP ?

????????????? cron crond ?????????? crontab ???????

2. ??????????????????????

- telnet : /etc/pam.d/remote (????????)
- SSH : /etc/pam.d/sshd (???????? SSHD)
- Local ???? : /etc/pam.d/login

????????????????????

```
# Limited users for remote login via telnet
# Check the file /etc/security/access.conf
account    required    pam_access.so
```

??????????

```
mkhomedir_helper <username>
```

??????????

?: ?????????????????????????????????

RedHat-KB: <https://access.redhat.com/solutions/65822>

```
# Create the restricted shell
cp /bin/bash /bin/rbash

# Create a directory that is used as the HOME of the user
mkdir /home/dbuser/
mkdir /home/dbuser/bin

# Modify the target user 'siview' for the shell as restricted shell
usermod -d /home/dbuser -s /bin/rbash siview

# or for new user
useradd -d /home/dbuser -s /bin/rbash siview
```

If a user uses **rbash**, the user can not do the following after login:

- Changing directories with the |cd| built in.
- Setting or unsetting the values of the |SHELL|, |PATH|, |ENV|, or |BASH_ENV| variables.
- Specifying command names containing slashes.
- Specifying a filename containing a slash as an argument to the |.| built in command.
- Importing function definitions from the shell environment at startup.
- Parsing the value of |SHELLOPTS| from the shell environment at startup.

- Redirecting output using the `>`, `>|`, `<>`, `>&`, `&>`, and `>>` redirection operators.
- Using the `|exec|` built in to replace the shell with another command.
- Adding or deleting built in commands with the `|-f|` and `|-d|` options to the `|enable|` built in.
- Specifying the `|-p|` option to the `|command|` built in.
- Turning off restricted mode with `|set +r|` or `|set +o restricted|`.

```
# Create specific profile for the user
vi /home/dbuser/.bash_profile
```

.bash_profile:

```
# cat /home/localuser/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi

# User specific environment and startup programs
PATH=$HOME/bin
export PATH
```

```
# Create the softlinks of commands which are required for the user
ln -s /bin/date /home/dbuser/bin/
ln -s /bin/ls /home/dbuser/bin/
ln -s /usr/bin/passwd /home/dbuser/bin/
```

????

- RH-KB: <https://access.redhat.com/solutions/66322> (RHEL6)
- RH-KB: [Set a password policy in Red Hat Enterprise Linux 7](#) (RHEL7)
- [How to Set password policy in CentOS or RHEL system](#)
- RedHat/CentOS: `/usr/share/doc/pam-<version>/txts/README.pam_cracklib`
- [??] <https://www.lijyjh.com/2012/07/pam-managing-account-security-with-pam.html>

????:

- difok=N , ?????? 5 ??
- minlen=N, ??????????? 9?

- dcredit=-1, 1 1 1 1
- ucredit=-1, 1 1 1 1 1 1 1
- lcredit=-1, 1 1 1 1 1 1 1

Edit `/etc/pam.d/system-auth` , `/etc/pam.d/password-auth`

CentOS 5/6)

NOTE: CentOS 5 ?? `/etc/pam.d/password-auth` , ???????
`/etc/pam.d/system-auth`

```
# Set password strength
#password requisite pam_cracklib.so try_first_pass retry=3 type=
password requisite pam_cracklib.so minlen=8 dcredit=-1 ucredit=-1 lcredit=-1
```

CentOS 7/8)

Edit `/etc/security/pwquality.conf`

```
# Set password strength
minlen = 8
dcredit = -1
ucredit = -1
lcredit = -1
```

?? root ?????????????????????? `/etc/pam.d/system-auth` ? `/etc/pam.d/password-auth` ??
password ?????? `enforce_for_root` ?

```
# Enforce root for password strength
password requisite pam_pwquality.so try_first_pass local_users_only retry=3
authtok_type= enforce_for_root
```

??????

Edit `/etc/pam.d/system-auth` , `/etc/pam.d/password-auth`

CentOS 5/6)

```
# Keep history of passwords used
# Add remember=N
# The last n passwords for each user are saved in /etc/security/opasswd in order to force
password change history
```

```
# and keep the user from alternating between the same password too frequently.
#password sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password sufficient pam_unix.so sha512 remember=8 shadow nullok try_first_pass
use_authtok
```

CentOS 7/8)

```
# Keep history of passwords used, insert the below line after pam_pwquality.so line
password requisite pam_pwhistory.so remember=8 use_authtok
```

“ TIP: ?????????? `/etc/security/opasswd` .

????

????????????????

```
# 1
grep ENCRYPT_METHOD /etc/login.defs

# 2
grep pam_unix.so /etc/pam.d/password-auth

# 3
passwd -S <user-name>
```

????????

```
authconfig --help | grep passalgo
```

??????

“ The below command modifies `/etc/pam.d/system-auth`, `/etc/login.defs`, `/etc/libuser.conf`, and `/etc/sysconfig/authconfig` and thus will have no effect (good or bad) on the password hashes already stored in `/etc/passwd` or `/etc/shadow`; instead, only future executions of `passwd` will use the new algo

```
authconfig --passalgo=sha512 --update
```

Optional: ?? non-root ?????????? <user-name>

```
for u in $(awk -F: '{if ( $1 != "root" && $2 ~ /^[!?:alnum:]\.\./ ) print $1}'  
/etc/shadow); do passwd --stdin $u <<<$u; done
```

????????????????????????????????

```
for u in $(awk -F: '{if ( $1 != "root" && $2 ~ /^[!?:alnum:]\.\./ ) print $1}'  
/etc/shadow); do chage -d0 $u; done
```

????

```
# Create a new group  
groupadd <group-name>  
addgroup <group-name>  
  
# add a group into an account  
usermod -aG mygroup user1  
useradd -aG family, friends james  
  
# To change the primary group of the user tom to family  
usermod -g family tom  
  
# remove user from a group  
gpasswd -d user1 mygroup  
  
# list all users in a group  
lid -g mygroup  
  
# list groups for current user  
groups  
  
# List groups for specified user  
groups username
```

?? passwd

```
# displays the status of user account password settings  
# [Username] [Status] [Date Last Changed] [Min. Age] [Max. Age] [Warn. Period] [ Inactivity  
Period]
```

```
# Status:
# - P: Usable password
# - NP: No password
# - L: Locked password
# Age:
# - 99999: Never expires
# - 0: Can be changed at anytime
# - -1: Disabled
passwd -S evans
evans PS 2020-09-07 0 99999 7 -1 (Password set, SHA512 crypt.)

# lock the password of a specified account
passwd -l user1

# unlock the password
passwd -u user2

# delete a password for an account
passwd -d user1

# expire a password for an account
# This will force user to change the password at next login.
passwd -e user2

# This sets the number of days before a password can be changed.
# By default, a value of zero is set, which indicates that the user may change
# their password at any time.
# This means user2 cannot change its own password until 10 days have passed.
passwd -n 10 user2

# To confirm the password setting made with the -n option above, run the following command:
# The value of 10 after the date indicates the minimum number of days
# until the password can be changed.
passwd -S user1
user1 PS 2020-12-04 10 99999 7 -1 (Password set, SHA512 crypt.)

# This means after 90 days, the password is required to be changed.
passwd -x 90 user2

# This means the user will receive warnings that the password will expire 7 days
```

```
# before the expiration.
passwd -w 7 user2

# This means after a user account has had an expired password for 5 days,
# the user may no longer sign on to the account.
passwd -i 5 user2

# This command will read from the echo command and pass it to the passwd command.
# So this will set the user1 password to userpasswd1.
echo "userpasswd1"|passwd --stdin user1
```

?? getent

```
# List all user
getent passwd
getent passwd | awk -F: '{print $1}'

# List a specified user
getent passwd <username>

# List the locked and no-login accounts
getent shadow | awk '/^.*:[!\*].*/' | cut -d: -f1

# List the users with uid between 1000 ~ 1500
getent passwd {1000..1500}
```

????????

```
# Step 1 – Create an encrypted password
## perl one liner ##
#perl -e 'print crypt("Your-Clear-Text-Password-Here", "salt"),"\n"'

password="1YelloDog@"
pass=$(perl -e 'print crypt($ARGV[0], "password")' $password)
echo "$pass"
```

```
# Step 2 – Shell script to add a user and password on Linux
#!/bin/bash
```

```

# Purpose - Script to add a user to Linux system including password
# Author - Vivek Gite <www.cyberciti.biz> under GPL v2.0+
# -----
# Am i Root user?
if [ $(id -u) -eq 0 ]; then
  read -p "Enter username : " username
  read -s -p "Enter password : " password
  egrep "^$username" /etc/passwd >/dev/null
  if [ $? -eq 0 ]; then
    echo "$username exists!"
    exit 1
  else
    pass=$(perl -e 'print crypt($ARGV[0], "password")' $password)
    useradd -m -p "$pass" "$username"
    [ $? -eq 0 ] && echo "User has been added to system!" || echo "Failed to add a user!"
  fi
else
  echo "Only root may add a user to the system."
  exit 2
fi

```

```

# Step 3 – Change existing Linux user’s password in one CLI
echo "vivek:password" | chpasswd

# Verify that password has been changed
chage -l vivek

```

```

# Step 4 – Create Users and change passwords with passwd on a CentOS/RHEL
echo "YourPassword" | passwd --stdin UserName

```

??????????

????????

```

# : uid=500
ID_minimum=500
for f in /etc/{passwd,group}; do awk -F: -vID=$ID_minimum '$3>=ID && $1!="nfsnobody"' $f |sort
-nt: -k3 > ${f#/etc/}.bak; done
while read line; do grep -w "^${line%:*}" /etc/shadow; done <passwd.bak >shadow.bak
while read line; do grep -w "^${line%:*}" /etc/gshadow; done <group.bak >gshadow.bak

```

```

# root: root uid=501 rootroot
export UGIDLIMIT=501
awk -v LIMIT=$UGIDLIMIT -F: '($3>=LIMIT) && ($3!=65534)' /etc/passwd | sed '/nfsnobody/d' >
passwd.move
awk -v LIMIT=$UGIDLIMIT -F: '($3>=LIMIT) && ($3!=65534)' /etc/group | sed '/nfsnobody/d' >
group.move
awk -v LIMIT=$UGIDLIMIT -F: '($3>=LIMIT) && ($3!=65534) {print $1}' /etc/passwd | egrep -wf -
/etc/shadow | sed '/nfsnobody/d' > shadow.move

# root: root uid=501 ~ 600 root
export UGID_DOWN=501
export UGID_UP=600
awk -v LIMIT_DOWN=$UGID_DOWN -v LIMIT_UP=$UGID_UP -F: '($3>=LIMIT_DOWN) && ($3<=LIMIT_UP) &&
($3!=65534)' /etc/passwd | sed '/nfsnobody/d' > passwd.move
awk -v LIMIT_DOWN=$UGID_DOWN -v LIMIT_UP=$UGID_UP -F: '($3>=LIMIT_DOWN) && ($3<=LIMIT_UP) &&
($3!=65534)' /etc/group | sed '/nfsnobody/d' > group.move
awk -v LIMIT_DOWN=$UGID_DOWN -v LIMIT_UP=$UGID_UP -F: '($3>=LIMIT_DOWN) && ($3<=LIMIT_UP) &&
($3!=65534) {print $1}' /etc/passwd | egrep -wf - /etc/shadow | sed '/nfsnobody/d' >
shadow.move

# root: uid= 501 ~ 699 and 1000+
export UGIDLIMIT_LOW=501
export UGIDLIMIT_HIGH=699
export UGIDS_RHEL7=1000

awk -v RHEL7=$UGIDS_RHEL7 -v LIMIT_LOW=$UGIDLIMIT_LOW -v LIMIT_HIGH=$UGIDLIMIT_HIGH -F:
'($3>=RHEL7) || ((($3>=LIMIT_LOW) && ($3<=LIMIT_HIGH) && ($3!=65534)))' /etc/passwd | sed
'/nfsnobody/d' > passwd.move

awk -v RHEL7=$UGIDS_RHEL7 -v LIMIT_LOW=$UGIDLIMIT_LOW -v LIMIT_HIGH=$UGIDLIMIT_HIGH -F:
'($3>=RHEL7) || ((($3>=LIMIT_LOW) && ($3<=LIMIT_HIGH) && ($3!=65534)))' /etc/group | sed
'/nfsnobody/d' > group.move

awk -v RHEL7=$UGIDS_RHEL7 -v LIMIT_LOW=$UGIDLIMIT_LOW -v LIMIT_HIGH=$UGIDLIMIT_HIGH -F:
'($3>=RHEL7) || ((($3>=LIMIT_LOW) && ($3<=LIMIT_HIGH) && ($3!=65534))) {print $1}' /etc/passwd |
egrep -f - /etc/shadow | sed '/nfsnobody/d' > shadow.move

```

🗨️ NOTE: ?????????????????????? /etc/gshadow ?????

?????? *.move ??????????????

```
cat passwd.move >> /etc/passwd
cat shadow.move >> /etc/shadow
cat group.move >> /etc/group
```

```
pwconv
grpconv
```

```
# ?????????? home ??????????
mkhomedir_helper <user-name>
```

Optional: ??????????

NOTE: ?????????? /etc/passwd ? /etc/group ?????? pwconv ?
grpconv ?????????? /etc/shadow ? /etc/gshadow ??????????????????

```
# ??????????
## ?? /etc/passwd
vipw

## ?? /etc/group
vigr

## ?? /etc/shadow, /etc/gshadow
pwconv
grpconv
```

Optional: ??? Home ??

```
for uidgid in $(cut -d: -f3,4 passwd.move); do
    dir=$(awk -F: /$uidgid/{print\$6} passwd.move)
    mkdir -vm700 "$dir"; cp -r /etc/skel/.[[[:alpha:]]* "$dir"
    chown -R $uidgid "$dir"; ls -ld "$dir"
done
```

?????? psacct

```
yum install psacct
```

- [How to Monitor Linux Users Activity with psacct or acct Tools](#)
- Display total statistics of connect time in hours
- Print All Linux Commands Executed by Users
- Print Linux User Information
- Print Number of Linux Processes
- Print and Sort Usage by Percentage
- Search Logs for Commands

???????? (TMOUT)

Linux: `/etc/profile.d/timeout.sh`

```
#!/bin/bash
# Set the TMOUT 600 for specified group
grpname="sshusers"
#if [[ "`id -Gn`" =~ .*"$grpname".* ]]; then
if grep -q "$grpname" <<< "`id -Gn`"; then
    export TMOUT=600
fi
```

Multi groups

```
#!/bin/bash
# Set the TMOUT 600 for specified groups
#grpnames="(group1|group2|group3)"
grpnames="(sshusers)"
if echo "`id -Gn`" | grep -wEq "$grpnames"; then
    export TMOUT=600
fi
```

AIX: `/etc/profile`

```
# Set the TMOUT 600 for specified groups
#grpnames="(group1|group2|group3)"
grpnames="(sshusers)"
if echo "`id -Gn`" | grep -wEq "$grpnames"; then
    export TMOUT=600
fi
```

Learning

- [How to Lock User Accounts After Failed Login Attempts](#)
- [Restrict SSH User Access to Certain Directory Using Chrooted Jail](#)
- [How can I restrict the normal user to run only limited set of commands in RHEL?](#)
- [How To Limit User's Access To The Linux System](#)
- [Set a password policy in Red Hat Enterprise Linux](#)
- [RedHat] [How to enhance Linux user security with Pluggable Authentication Module settings](#)
- [Linux PAM for Compliance](#)
- [12 Ways to Find User Account Info and Login Details in Linux](#)

Revision #114

Created 2020-10-21 07:15:51 CST by A-Lang (Admin)

Updated 2026-01-13 15:37:01 CST by A-Lang (Admin)