User Accounts Management

1. Creating a group

Create a group "osadmins":

```
# groupadd osadmins
```

Set group members full system administration privileges:

```
# echo "%osadmins ALL=(ALL) ALL" > /etc/sudoers.d/osadmins
```

---

2. Creating a user

Create a user "admin1" and assign it to the existing group "osadmins":

```
# useradd -G osadmins admin1
```

Add the user's own name "Jan Novak":

```
# usermod -c "Jan Novak" admin1
```

Set a password for the user:

```
# passwd admin1
```

Make sure the user changes the password on the first login:

```
# chage -d 0 admin1
```

---

3. Creating multiple users

Create users "admin01–05" and assign them the specified passwords:

```
# echo "admin01 kYT97kk16r41cKU
admin02 pWm69Sj98J92Zvm
admin03 pWL74qU72S59sEV
admin04 rNp29Zp81P22XZi
```

```
admin05 fiq64WC24a89CvW" | while read usr pswd; do useradd $usr; echo $pswd
| passwd --stdin $usr; done
```

Set a maximum password validity of 90 days for the users:

```
# for usr in admin0{1..5}; do chage -M 90 $usr; echo $usr; chage -l $usr |
grep Maximum; done
```

---

4. Locking and unlocking a user

Lock/unlock or verify the status of the "admin1" user account:

```
# passwd -l admin1
# passwd -u admin1
# passwd -S admin1
```

or

```
# usermod -L admin1
# usermod -U admin1
# passwd -S admin1
```

or

```
# chsh -s /bin/false admin1
# chsh -s /bin/bash admin1
# awk -F ":" '/admin1/ {print $7}' /etc/passwd
```

or

```
# chage -E 0 admin1
# chage -E -1 admin1
# chage -l admin1
```

Display or reset the number of failed logins of the user "admin1":

```
# pam_tally2 -u admin1
# pam_tally2 --reset -u admin1
```

or

```
# faillock --user admin1
# faillock --user admin1 --reset
```

or

```
# faillog -u admin1
# faillog -u admin1 -r
```

---

5. Removing a user

Remove a user "admin1" including its home directory and email spool:

```
# userdel -r admin1
```

---

6. Securing user accounts

Display duplicate UID:

```
# duid=$(awk -F ":" '{print $3}' /etc/passwd | sort | uniq -d); [[ -z "$duid" ]] && echo "NONE" || egrep "$duid:[0-9]" /etc/passwd
```

Display duplicate GID:

```
# dgid=$(awk -F ":" '{print $3}' /etc/group | sort | uniq -d); [[ -z "$dgid" ]] && echo "NONE" || egrep "$dgid" /etc/group
```

Display default "PASS_MIN_LEN" value:

```
# echo $(sed '/^[^#]*PASS_MIN_LEN/!d' /etc/login.defs | awk '{print $2}' 2> /dev/null); [[ $? -ne 1 ]] && echo "" $(sed '/^[^#]/!d' /etc/pam.d/common-password /etc/security/pam_pwcheck.conf 2> /dev/null | egrep -o "minlen=." | awk -F "=" '{print $2}')
```

Display default "PASS_MIN_DAYS" value:

```
# echo $(sed '/^[^#]*PASS_MIN_DAYS/!d' /etc/login.defs)
```

Display default "PASS_MAX_DAYS" value:

```
# echo $(sed '/^[^#]*PASS_MAX_DAYS/!d' /etc/login.defs)
```

Display current "PASS_MIN_DAYS" and "PASS_MAX_DAYS" values for users with a password assigned:

```
# pa=$(awk -F ":" '{if (length($2) > 2) print $1}' /etc/shadow); for usr in "$pa"; do egrep "^$usr" /etc/shadow | awk -F ":" '{printf "%-15s %-15s %-15s\n", $1, $4, $5}'; done
```

Display "password history" value:

```
# echo $(sed '/^[^#]/!d' /etc/pam.d/system-auth /etc/pam.d/common-password
/etc/security/pam_pwcheck.conf 2> /dev/null | egrep -o "remember=.")
```

Display "loginretries" value:

```
# echo $(lr=$(sed '/^[^#]/!d' /etc/pam.d/system-auth /etc/pam.d/common-auth
/etc/pam.d/common-account 2> /dev/null | egrep -o "deny=."); [[ -n "$lr" ]]
&& echo "$lr" || sed '/^[^#]*LOGIN_RETRIES/!d' /etc/login.defs 2> /dev/null
| awk '{print $2}')
```

Display users with empty passwords:

```
# for pwd in $(awk -F ":" '{ print $2 }' /etc/shadow); do if [[ "${#pwd}" -
lt 1 ]]; then grep "$pwd" /etc/shadow; fi; done; [[ "${#pwd}" -gt 0 ]] &&
echo "No empty password assigned."
```

Display users with never expiring passwords:

```
# ne=$(awk -F ":" '{if (length($2) > 2 && ($5 == "" || $5 >= 99999)) print
$1}' /etc/shadow); [[ -z "$ne" ]] && echo "NONE" || echo "$ne" | tr " " "\n"
```

Display users with locked accounts:

```
# la=$(for user in $(awk -F ":" '{if ($7 ~ /.+(sh|bash|ksh|zsh)$/ && ($3 ==
0 || $3 >= 500)) print $1}' /etc/passwd); do passwd -S $user | grep "LK";
done); [[ -z "$la" ]] && echo "NONE" || echo "$la"
```

Display the password encrypt method:

```
# echo $(sed '/^[^#]*\(sha[25]\|md5\)/!d' /etc/pam.d/system-auth
/etc/pam.d/common-password 2> /dev/null | egrep -o "sha[1256]{3}|md5") ||
sed '/^[^#]*ENCRYPT_METHOD/!d' /etc/login.def
```

Display users whose passwords are stored in /etc/passwd:

```
# pp=$(awk -F ":" '{if (length($2) > 2) print $1}' /etc/passwd); [[ -z "$pp"
]] && echo "NONE" || echo "$pp" | tr " " "\n"
```

Display default "umask" value:

```
# echo $(redhat=$(sed '/^[^#]*umask/!d;q' /etc/bashrc 2> /dev/null | awk
'{print $2}'); suse=$(sed '/^[^#]*umask/!d' /etc/profile.local 2> /dev/null
```

```
| awk '{print $2}'); ubuntu=$(sed '/^[^#]*UMASK/!d' /etc/login.defs 2>
/dev/null | awk '{print $2}'); if [[ -n "$redhat" ]]; then echo "$redhat";
elif [[ -n "$suse" ]]; then echo "$suse"; else echo "$ubuntu"; fi)
```

Display default "umask" for home directories:

```
# echo $(sed '/^[^#]*UMASK/!d' /etc/login.defs 2> /dev/null | awk '{print
$2}')
```

Display home directories with permissions different from value of "700":

```
# homedir=$(awk -F ":" '{if ($7 ~ /.+(sh|bash|ksh|zsh)$/ && ($3 == 0 || $3
>= 500)) print $6}' /etc/passwd | xargs ls -ld 2> /dev/null | egrep -v
"^drwx------"); [[ -z "$homedir" ]] && echo "NONE" || echo "$homedir"
```

Display home directories with non-standard ownership:

```
# io=$(for a in $(awk -F ":" '{if ($7 ~ /.+(sh|bash|ksh|zsh)$/ && ($3 == 0
|| $3 >= 500)) print $6}' /etc/passwd); do b=$(stat -c %U $a 2> /dev/null);
c=$(stat -c %G $a 2> /dev/null); d=$(echo $a | awk -F "/" '{print $NF}'); if
[[ "$b" != "$d" || "$c" != $(id -ng "$b") ]]; then ls -ld "$a"; fi; done);
[[ -z "$io" ]] && echo "NONE" || echo "$io"
```