VERSI 1.0

FEBRUARI, 2023

PRAKTIKUM SISTEM OPERASI

Managing Local User and Group

TIM PENYUSUN:

MAHAR FAIQURAHMAN, S.KOM., M.T.
MUHAMMAD RIDHA AGAM
SYAHRUL PANGESTU

PRESENTED BY: LAB. INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

MODUL [1] SISTEM OPERASI – MANAGING LOCAL USER AND GROUP

CAPAIAN PEMBELAJARAN MATA KULIAH

- Dapat mendeskripsikan fungsi dan kegunaan *users* dan *groups* pada sistem linux
- Dapat berpindah menjadi akun superuser untuk mengelola sebuah linux
- Dapat membuat, memodifikasi, dan menghapus akun lokal user.
- Dapat membuat, memodifikasi, dan menghapus akun lokal group.
- Dapat mensetting aturan password manajemen untuk user, dan lock atau unlock akun user.

KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/PC
- Virtual Machine (VMware, Virtual Box, VPS bila mampu ehe :")
- Sistem Operasi CentOS, download image OVA (Wajib)

MATERI POKOK

- Describing User and Group Concepts
- Gaining Superuser Access
- Managing Local User Accounts
- Managing Local Group Accounts
- Managing User Passwords

MATERI PRAKTIKUM

1. Describing User and Group Concepts

User

Akun *user* digunakan untuk menyediakan batasan keamanan antara orang satu dengan yang lain dan program yang dapat menjalankan perintah (*command*).

Semua *user* mempunyai *username* untuk mengidentifikasi *user* itu sendiri dan membuat lebih mudah digunakan. Masing-masing akun *user* dibedakan dengan *unique identification number* yang telah didefinisikan di dalam sistem, yang mana disebut dengan *User ID* (UID). Secara umum, akun *user* dilengkapi dengan *password* untuk nantinya digunakan untuk membuktikan bahwa *user* tersebut benar-benar terotorisasi atau sah ketika melakukan *login in*.

Seperti yang sempat disinggung di awal, bahwa akun user merupakan hal fundamental pada sistem keamanan. Setiap proses (program yang berjalan) pada sistem dijalankan oleh *user* tertentu dan setiap file dimiliki oleh *user* tertentu sebagai *owner.* Kepemilikan *file* membantu sistem menjalankan akses kontrol *user* terhadap *file*. Sehingga dapat disimpulkan *user* berkaitan dengan proses yang berjalan, menentukan file dan direktori yang dapat diakses oleh proses tersebut.

Terdapat tiga tipe akun *user* diantaranya: *superuser, system users, regular users.*

- Akun *superuser* digunakan untuk administrasi sistem. Nama lain dari *superuser* adalah **root** dan mempunyai akun UID 0. Superuser mempunyai semua akses terhadap sistem.

- Sistem mempunyai akun *system user,* digunakan oleh proses yang membutuhkan layanan support (*supporting services*). Proses atau *daemon* (proses yang berjalan di background) ini biasanya tidak perlu dijalankan sebagai *superuser.* Mereka diberikan akun *non-privileged* yang memungkinkan untuk mengamankan *file* dan *resource* satu sama lain dan mengamankan *file, resource* dari *regular user* yang ada pada sistem.
- Kebanyakan *user* mempunyai akun *regular user,* digunakan untuk pekerjaan sehari-hari jika tidak membutuhkan *superuser*. Seperti *system users, regular users* mempunyai akses yang terbatas terhadap sistem.

Kita juga dapat menampilkan informasi mengenai user yang sedang *logged-in* dengan menggunakan perintah **id**.

```
[user01@host ~]$ id
uid=1000(user01) gid=1000(user01) groups=1000(user01)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Untuk melihat *basic information* mengenai *user* lain, masukan *username* milik *user* yang ingin dilihat ke *command* **id** sebagai argumen.

```
[user01@host]$ id user02
uid=1002(user02) gid=1001(user02) groups=1001(user02)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Untuk melihat pemilik (*owner*) *file* gunakan command **ls -l**. Sedangkan jika ingin melihat kepemilikan direktori gunakan *command* **ls -ld**. Pada *output* command berikut, kolom ketiga menunjukan username.

```
[user01@host ~]$ ls -l file1
-rw-rw-r--. 1 user01 user01 0 Feb  5 11:10 file1
[user01@host]$ ls -ld dir1
drwxrwxr-x. 2 user01 user01 6 Feb  5 11:10 dir1
```

Untuk melihat informasi proses, gunakan *command* **ps**. Secara *default* hanya untuk menunjukan proses yang berjalan pada shell saat ini. Tambahkan opsi **a** untuk melihat semua proses dengan terminal. Untuk melihat proses yang dijalankan oleh *user* lain, masukan opsi **u**. Pada *output* berikut ini kolom pertama merupakan *username.*

```
[user01@host]$ ps -au
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       777  0.0  0.0 225752  1496 tty1     Ss+  11:03   0:00 /sbin/agetty -o -
p -- \u --noclear tty1 linux
root       780  0.0  0.1 225392  2064 ttyS0    Ss+  11:03   0:00 /sbin/agetty -o -
p -- \u --keep-baud 115200,38400,9600
user01      1207  0.0  0.2 234044  5104 pts/0    Ss   11:09   0:00 -bash
user01      1319  0.0  0.2 266904  3876 pts/0    R+   11:33   0:00 ps au
```

Output yang ditampilkan *command* diatas menampilkan *users* dengan username, walaupun ketika di dalam internal sistem operasi menggunakan UID untuk melacak *users*. Pemetaan *username* ke UID telah didefinisikan di dalam database *account information*. Secara *default,* sistem menggunakan file **/etc/passwd** untuk menampung informasi mengenai *local users.*

Setiap baris pada file **/etc/passwd** memuat informasi mengenai sebuah *user.* Setiap baris dipisahkan menjadi tujuh bagian dengan dipisahkan dengan titik dua. Berikut adalah contoh salah satu baris dari isi yang terdapat pada file **/etc/passwd**:

```
❶user01:❷x:❸1000:❹1000:❺User One:❻/home/user01:❼/bin/bash
```

1. *Username user* kali ini adalah (**user01**).
2. *Password user* terletak pada bagian ini dengan format yang telah dienkripsi. Bagian ini selalu dituliskan (**x**).
3. Nomor UID akun *user* (**1000**)
4. Nomor GID (***Group* ID**) atau *group* utama *user* (**1000**). *Group* akan dijelaskan nanti pada modul ini.
5. Nama asli *user* (**User One**)
6. *Home directory* untuk *user* (**/home/user01**). Ini merupakan *initial working directory* ketika program shell berjalan dan memuat data *user* beserta *configuration settings*-nya.
7. *Default* program *shell* untuk *user*, berjalan pada login (**/bin/bash**). Untuk *regular user*, ini merupakan program yang menyediakan *command-line prompt* milik *user*. *System user* dapat menggunakan **/sbin/nologin** jika *user* tersebut tidak diizinkan untuk *login*.

Group

Group adalah kumpulan *user* untuk membagi akses terhadap *file* dan sumber daya sistem lainnya. *Group* dapat digunakan untuk membagi akses *file* ke semua *user* yang masuk ke dalam *group* tersebut daripada membagi satu-satu ke *user*.

Seperti *user,* Group mempunyai nama *group* agar mudah dipakai. Secara internal, sistem membedakan grup berdasarkan *unique identification number* yang telah diberikan atau biasa disebut *Group* ID (GID).

Pemetaan nama *group* ke GID didefinisikan di dalam *database group account information*. Secara *default*, sistem menggunakan *file* **/etc/group** untuk menampung informasi *local group*.

Setiap baris di dalam *file* **/etc/group** memuat informasi mengenai suatu *group*. Setiap *group* yang ada pada *file* tersebut dibagi menjadi empat bagian dipisahkan dengan titik dua. Berikut adalah salah satu contoh baris di dalam **/etc/group**:

```
❶group01:❷x:❸10000:❹user01,user02,user03
```

1. Nama *group* ini (**group01**).
2. Bagian *password group*. Bagian ini selalu diisi **x**.
3. GID *group* ini adalah (**1000**).
4. List *user* yang menjadi anggota dari *group* sebagai *supplementary group* (**user01, user02, user03**). *Primary* (*default*) dan *supplementary group* akan dijelaskan pada modul ini.

Primary Group and Supplementary Group

Secara *default*, setiap *user* pasti mempunyai sebuah *primary group*. Di dalam *local user*, primary group lah yang terdaftar pada *file* **/etc/passwd** dengan nomor GID-nya. Secara *default*, *group* juga akan memiliki *file* baru yang telah dibuat *user*.

Normalnya, ketika kita membuat sebuah *regular user* baru, sebuah *group* baru dengan nama yang sama juga akan terbuat. *group* ini digunakan sebagai *primary group* oleh *user* yang baru dan hanya *user* tersebut sebagai anggota *User Private Group* ini. Hal seperti ini dapat membantu pengelolaan *file permission* lebih sederhana, untuk lebih jelasnya akan dibahas setelah ini.

User mungkin juga mempunyai *supplementary group*. Keanggotaan pada *supplementary group* tercantum pada file ***/etc/group***. *User* diberikan akses terhadap *file* berdasarkan kumpulan *group* yang dimiliki oleh *user* mempunyai akses terhadap *file* tersebut atau tidak, entah *group* tersebut memiliki tipe *primary* atau *supplementary*.

Contohnya jika *user* **user01** mempunyai *primary group* **user01** dan memiliki *supplementary group* *wheel* dan *webadmin*, maka *user* dapat membaca *readable file* yang dimiliki ketiga *group* tersebut.

Untuk menemukan informasi mengenai anggota *group*, gunakan *command* **id**.

```
[user03@host ~]$ id
uid=1003(user03) gid=1003(user03) groups=1003(user03),10(wheel),10000(group01)
 context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Pada contoh di atas, **user03** mempunyai *group* **user03** sebagai *primary group* (GID). *Item group* milik *user* ini sudah tercantum semua, selain *primary group* **user03**, *user* ini juga mempunyai *supplementary group* ***wheel*** dan **group01**.
Quiz

2. Gaining Superuser Access

Superuser

Kebanyakan sistem operasi mempunyai semacam *superuser*, sebuah *superuser* mempunyai kekuasaan atas sistem. Pada Red Hat Linux biasa disebut *user* **root**. *User* ini mempunyai kuasa untuk meng-*override normal privileges* pada *file system*, dan digunakan untuk mengelola sistem. Jika melaksanakan pekerjaan seperti *install* atau *remove software* dan untuk mengelola *file* sistem beserta direktori, *user* harus menaikan *privilege* sebagai *user* **root**.

Singkatnya, *user* **root** merupakan *user* normal yang dapat mengontrol sebagian besar perangkat, namun terdapat beberapa perbedaan dengan user normal. Contohnya, normal *user* dapat mengontrol *removable* devices seperti USB. Normal *user* tersebut dapat menambah, menghapus *file* dan mengelola USB tersebut, sedangkan **root** dapat mengelola *hard drive* "*fixed*" secara *default*.

Seperti yang dikatakan bibi may, "*with great power there must also come great responsibility*". *User* **root** memiliki *power* yang OP untuk merusak sistem seperti: menghapus *file* dan direktori, menghapus akun *user*, menambahkan *back doors*, dan lain-lain masih banyak sekali :". Jika *user* **root** disebar, orang lain mungkin akan mengambil alih kontrol sistem. Pada modul ini, *administrator* (praktikan) disarankan untuk *login* sebagai normal user untuk nantinya menaikan *privilege* ke **root** hanya ketika dibutuhkan.

Akun **root** pada Linux mungkin setara dengan akun *local administration* di Microsoft Windows. Pada Linux, kebanyakan sistem *administrator login* ke dalam sistem sebagai *unprivileged user* dan menggunakan berbagai *tool* untuk meningkatkan *privilege* **root** sementara.

Switching Users

Command **su** memungkinkan *user* berpindah ke akun *user* yang berbeda. Jika kita menjalankan *command* **su** dari *regular user*, kita akan memasukan *password* akun mana yang diinginkan. Ketika **root** menjalankan **su**, maka tidak perlu memasukan *password* lagi.

```
[user01@host ~]$ su - user02
Password:
[user02@host ~]$
```

Jika kita lupa *username*, gunakan *command* **su -** atau **su** saja untuk berganti ke **root** secara *default*.

```
[user01@host ~]$ su -
Password:
[root@host ~]#
```

Command **su** memulai *non-login shell*, ketika *command* **su -** (dengan *dash*) maka *login shell* akan berjalan. Perbedaan utama antara dua *command* tersebut adalah **su -** mengatur *environment shell* seolah-olah menjadi *login user* baru, sedangkan **su** hanya memulai *shell* sebagai *user*, tetapi menggunakan o*riginal environment setting* milik *user* sendiri.

Dalam kebanyakan kasus, *administrator* harus menjalankan **su -** untuk mendapatkan *shell* dengan target *environment setting* normal milik *user*.

Running Command with Sudo

Dalam beberapa kasus, akun **root** milik *user* mungkin masih belum mempunyai sebuah *password* yang *valid* untuk alasan keamanan. Dalam kasus ini, *user* tidak dapat *login* kedalam sistem sebagai **root** secara langsung dengan *password*, dan *command* **su** tidak dapat digunakan untuk mendapatkan *interactive shell*. Sebuah *tool* yang dapat digunakan untuk mendapatkan akses *root* dalam kasus ini adalah **sudo**.

Berbeda dengan **su**, **sudo** normalnya membutuhkan *user* untuk memasukan *password* milik mereka sendiri agar dapat melakukan autentikasi, bukan *password* akun *user* yang ingin diakses. Sehingga *user* yang menggunakan **sudo** sebagai **root** tidak perlu mengetahui **root** *password user lain* untuk menjalankan *command*. Lebih baik mereka menggunakan *password* mereka sendiri untuk mengautentikasi akses.

Sebagai tambahan, **sudo** dapat dikonfigurasi agar dapat memberikan izin kepada spesifik *user* untuk menjalankan *command* sebagaimana *user* lain, atau hanya beberapa *command* seperti *user* tertentu.

Sebagai contoh, ketika **sudo** dikonfigurasi untuk mengizinkan **user01** menjalankan *command* **usermod** sebagai **root**, **user01** dapat menjalankan *command* berikut ini untuk *lock* atau *unlock* akun *user* lain:

```
[user01@host ~]$ sudo usermod -L user02
[sudo] password for user01:
[user01@host ~]$ su - user02
Password:
su: Authentication failure
[user01@host ~]$
```

Jika sebuah akun mencoba menjalankan *command* sebagai *user* lain, dan **sudo** *configuration* tidak diizinkan, maka *command* akan diblokir, percobaan akan masuk kedalam *log*, dan secara *default email* akan dikirimkan ke *user* **root**.

```
[user02@host ~]$ sudo tail /var/log/secure
[sudo] password for user02:
user02 is not in the sudoers file.  This incident will be reported.
[user02@host ~]$
```

Sebuah keuntungan tambahan menggunakan **sudo** yaitu, semua *command* yang telah dieksekusi tercatat secara *default* ke **/var/log/secure**.

```
[user01@host ~]$ sudo tail /var/log/secure
...output omitted...
Feb  6 20:45:46 host sudo[2577]:  user01 : TTY=pts/0 ; PWD=/home/user01 ;
 USER=root ; COMMAND=/sbin/usermod -L user02
...output omitted...
```

Pada Red Hat *Enterprise* Linux 7 dan Red Hat *Enterprise* 8, semua *member* dari *group* **wheel** dapat menggunakan **sudo** untuk menjalankan *command* seperti *user* lain, termasuk **root**. *User* diarahkan ke *pasword* mereka masing-masing. Ini adalah sebuah perubahan dari Red Hat *Enterprise* Linux 6 dan yang lebih dulu, dimana *user* yang menjadi anggota *group* **wheel** tidak mendapatkan akses administratif ini secara *default*.

Getting an Interactive Root Shell with Sudo

Jika terdapat akun *user non-administrative* yang dapat menggunakan **sudo** untuk menjalankan command **su**, kita dapat menjalankan **sudo su -** dari akun tersebut untuk mendapatkan *interactive* **root** *user shell*. Ini dapat dijalankan karena **sudo** akan menjalankan **su -** sebagai **root**, dan **root** tidak memerlukan *password* lagi jika menggunakan **su**.

Cara lain untuk mengakses akun **root** dengan **sudo** adalah menggunakan command **sudo -i**. Ini akan memindah menjadi akun **root** dan menjalankan shell user tersebut secara *default* (biasanya bash) dan terkait dengan *shell login script*. Jika kita hanya ingin menjalankan *shell*, kita dapat menggunakan *command* **sudo -s**.

Contohnya, seorang *administrator* mungkin mendapatkan sebuah *interactive shell* sebagai *root* pada AWS EC2 *instance* dengan menggunakan SSH *public-key authentication* untuk *login* sebagai *user* normal uc2-user, dan selanjutnya menjalankan **sudo -i** untuk mendapatkan **user shell root**.

```
[ec2-user@host ~]$ sudo -i
[sudo] password for ec2-user:
[root@host ~]#
```

3. Managing Local User Accounts

Berikut beberapa *command-line tools* yang dapat digunakan untuk mengelola akun lokal *user*.

Creating users from the command line

- *Command* **useradd username** membuat sebuah *user* baru bernama *username*. Dimana *user* ini dapat mengolah home directory dan account information milik user tersebut, membuat *private group* untuk *user* dengan nama *username*. Pada tahap ini, akun belum mempunyai *password* yang *valid*, dan *user* tidak dapat *login* sebelum *setting password*.
- *Command* **useradd –help** menampilkan opsi *basic*. Pada banyak kasus, opsi yang sama dapat digunakan dengan *command* **usermod** untuk memodifikasi *user* yang ada.
- Beberapa *default*, seperti *range* nomor UID yang *valid*, aturan *default* umur *password* dibaca dari *file* **/etc/login.defs**. *Value* pada *file* ini hanya digunakan ketika membuat *user* baru. Sebuah perubahan pada *file* ini tidak mempengaruhi *user* yang ada.

Modify Existing Users from the Command Line

- *Command* **usermod –help** menampilkan opsi *basic* yang dapat digunakan untuk memodifikasi sebuah akun. Beberapa opsi termasuk:

Usermod Option	Penggunaan
-c, –comment COMMENT	Menambahkan nama asli milik *user* pada kolom *comment.*
-g, –gid GROUP	Melihat *primary group* akun *user.*
-G, –groups GROUPS	Melihat daftar *supplementary group* akun pengguna.
-a, –append	Untuk menambah keanggotaan *supplementary group* kepada *user.*
-d, –home HOME_DIR	Melihat *home directory* khusus akun *user.*
-m, –move-home	Memindah *home directory* milik akun *user* ke lokasi yang baru. Harus digunakan dengan opsi -d.
-s, –shell SHELL	Melihat *login shell* akun *user* tertentu.
-L, –lock	*Lock* akun *user*.
-u, –unlock	*Unlock* akun *user*

Deleting Users from the Command Line

- *Command* **userdel username** menghapus *detail username* dari **/etc/passwd**, tetapi masih menyisakan *home directory* milik *user*.

- *User* **root** dapat memberikan sebuah *password* kepada *user* dengan nilai apapun. Pesan akan ditampilkan secara langsung jika *password* tidak memenuhi standar minimum yang dianjurkan, dan akan diarahkan untuk mengisi password yang baru lagi dan semua token akan diupdate jika sudah berhasil.

```
[root@host ~]# passwd user01
Changing password for user user01.
New password: redhat
BAD PASSWORD: The password fails the dictionary check - it is based on a
 dictionary word
Retype new password: redhat
passwd: all authentication tokens updated successfully.
[root@host ~]#
```

- Sebuah *regular user* harus membuat sebuah *password* dengan spesifikasi minimal sepanjang delapan karakter dan bukan kata-kata dari kamus, *username* atau bahkan *password* sebelumnya.

UID Ranges

Nomor UID yang spesifik dan *range* nomor tersebut digunakan untuk tujuan tertentu oleh Red Hat *Enterprise* Linux.

- UID 0 selalu diberikan untuk akun *superuser* **root**.
- UID 201-999 adalah *range* UID "*system users*"
- UID 1-200 adalah *range* UID "*system users*", digunakan oleh proses sistem yang tidak memiliki *file* pada *file system*. *range* tersebut biasanya didefinisikan secara dinamis dari *available pool* ketika *software* perlu diinstall. Program yang berjalan disini sebagai "*unprivileged*" *system user* untuk membatasi akses mereka agar hanya dapat mengakses *resource* yang mereka butuhkan untuk *function*.
- UID 1000+ merupakan *range* yang tersedia untuk diberikan kepada *regular user*.

4. Managing Local Group Accounts

Managing Local Groups

Group harus ada sebelum *user* dapat dimasukan kedalam sebuah *grup*. Beberapa *command-line tools* yang digunakan untuk mengelola *local group account*

Creating Groups from the Command-Line

- *Command* **groupadd** membuat *group*. Ketika membuat *group*, *command* **groupadd** menggunakan GID selanjutnya yang tersedia dari *range* yang sudah didefinisikan di dalam *file* **/etc/login.defs**.
- Opsi **-g** menentukan GID tertentu untuk digunakan oleh *group*.

```
[user01@host ~]$ sudo groupadd -g 10000 group01
[user01@host ~]$ tail /etc/group
...output omitted...
group01:x:10000:
```

- Opsi **-r** membuat sebuah *system group* menggunakan sebuah GID dari range GID *system* yang valid dan sudah di-*list* pada *file* **/etc/login.defs**. *Item* konfigurasi SYS_GID_MIN dan SYS_GID_MAX di dalam file **/etc/login.defs** mendefinisikan *range* dari sistem GID.

```
[user01@host ~]$ sudo groupadd -r group02
[user01@host ~]$ tail /etc/group
...output omitted...
group01:x:10000:
group02:x:988:
```

Modifying Existing Groups from the Command Line

- *Command* **groupmod** merubah *properties gorup* yang ada. Opsi **-n** digunakan untuk menentukan nama *group* baru.

```
[user01@host ~]$ sudo groupmod -n group0022 group02
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:988:
```

Bisa dilihat, nama *group* di atas sudah ter-*update* dari **group02** menjadi **group0022**.

- Opsi **-g** untuk menentukan GID yang baru

```
[user01@host ~]$ sudo groupmod -g 20000 group0022
[user01@host ~]$ tail /etc/group
...output omitted...
group0022:x:20000:
```

GID sudah ter-*update* dari 988 menjadi 20000

Deleting Groups from the Command Line

- *Command* **groupdel** menghapus *group*

```
[user01@host ~]$ sudo groupdel group0022
```

Changing Group Membership from the Command Line

- Keanggotaan *group* dikontrol oleh *user management*. Gunakan *command* **usermod -g** untuk merubah *primary group user*.

```
[user01@host ~]$ id user02
uid=1006(user02) gid=1008(user02) groups=1008(user02)
[user01@host ~]$ sudo usermod -g group01 user02
[user01@host ~]$ id user02
uid=1006(user02) gid=10000(group01) groups=10000(group01)
```

- Gunakan *command* **usermod -aG** untuk menambah *user* kedalam *supplementary group*.

```
[user01@host ~]$ id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03)
[user01@host ~]$ sudo usermod -aG group01 user03
[user01@host ~]$ id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(group01)
```

5. Managing User Passwords

Shadow Passwords and Password Policy

Dalam satu waktu, *password* yang terenkripsi ditampung pada file **/etc/passwd**. Hal ini dianggap aman hingga akhirnya *dictionary attack* terhadap *password* yang terenkripsi menjadi marak. Karena poin tersebut, *password* yang terenkripsi dipindah ke *file* terpisah **/etc/shadow** yang

mana hanya *readable* oleh **root** saja. *File* **/etc/shadow** juga memungkinkan untuk mengatur jangka *expired* fitur yang sudah diimplementasikan.

Seperti **/etc/passwd**, setiap *user* mempunyai baris di dalam *file* **/etc/passwd**. Sebuah sampel dari salah satu bari di dalam *file* **/etc/shadow** dengan dipisahkan menjadi sembilan bagian yang dipisahkan oleh titik dua seperti berikut ini:

```
❶user03:❷$6$CSsX...output omitted...:❸17933:❹0:❺99999:❻7:❼2:❽18113:❾
```

1. *Username* pemilik *password* tersebut.
2. *Password* yang terenkripsi milik *user*. Format enkripsi *password* akan dibahas nanti pada modul ini.
3. Hari dimana *password* terakhir diganti. Contoh diatas menunjukan terakhir di*update* pada 1970-01-01, dan dikalkulasi menggunakan UTC *time zone*.
4. Jumlah minimal hari yang harus dilalui semenjak *password* terakhir diubah, sebelum *user* dapat mengubahnya lagi.
5. Jumlah maksimal hari dapat lolos menggunakan *password* yang belum diubah, sebelum *password* tersebut *expired*. Jika *field* kosong berarti tidak akan *expired* dari hari terakhir diubah.
6. Periode *warning*. *User* akan diperingatkan bahwa *password* akan kadaluarsa. Warning akan muncul ketika mereka *login* menginjak jumlah hari yang diberikan pada *warning period* sebelum *deadline* tiba.
7. Periode inaktif. Setelah *expired*, password akan tetap diterima untuk *login* selama beberapa hari. Setelah jumlah *periode* warning ini berlalu, akun akan terblokir.
8. Hari dimana akun *expired*. Disini disetting dari 1970-01-01, dan dikalkulasi menggunakan UTC *time zone*. *Field* yang kosong berarti tidak kadaluarsa pada tanggal tertentu.
9. *Field* terakhir biasanya kosong dan dicadangkan untuk penggunaan di masa mendatang

Format of an Encrypted Password

Field password yang terenkripsi ditampung pada tiga bagian dari informasi: algoritma *hashing* yang digunakan, *salt*, dan *hash* yang dienkripsi. Setiap bagian informasi dibatasi oleh tanda $.

```
$❶6$❷CSsXcYG1L/4ZfHr/$❸2W6evvJahUfzfHpc9X.45Jc6H30E...output omitted...
```

1. Algoritma *hashing* digunakan untuk *password* ini. Nomor 6 mengindikasikan menggunakan hash SHA-512, yang mana merupakan *default* pada Red Hat *Enterprise* Linux 8. 1 dapat mengindikasikan MD5, 5 mengindikasikan SHA-256.
2. *Salt* digunakan untuk mengenkripsi *password*. Kombinasi karakter dipilih dengan *random*.
3. *Hash* yang terenkripsi dari *password* pengguna. *Salt* dan *password* belum terenkripsi dikombinasikan dan dienkripsi untuk menghasilkan *hash* yang terenkripsi dari *password*.

Alasan utama mengkombinasi *salt* dengan *password* adalah untuk mempertahankan dari serangan menggunakan *precomputed list hash password*. Menambahkan *salt* dapat mengubah hasil dari *hash*, sehingga membuat *precomputed list* tidak berguna. Jika *attacker* berhasil meng-*copy file* **/etc/shadow** yang menggunakan *salt*, mereka tidak bisa melakukan *brute force password* untuk menebak, karena sangat memakan waktu dan *effort*.

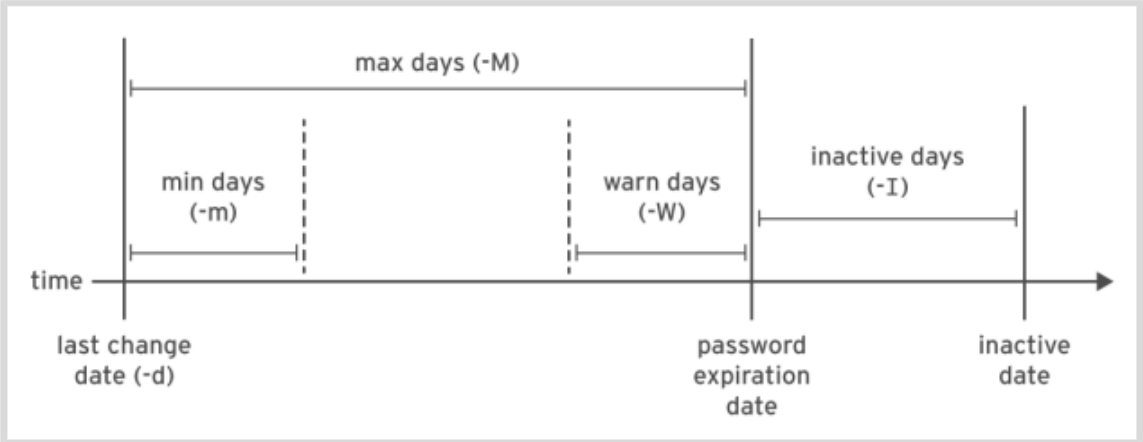
Password Verification

Ketika *user* mencoba *login*, sistem mengecek *user* yang masuk di dalam file **/etc/shadow**, mengkombinasikan *salt* dengan *unencrypted password* yang dimasukan *user*, setelah itu

mengenkripsinya menggunakan *hashing algorithm* yang sudah ditentukan. Jika hasilnya sama, maka *user* memasukan *password* dengan benar. Jika hasilnya tidak sama dengan *encrypted hash*, maka *user* memasukan *password* yang salah dan gagal *login*. Metode ini memungkinkan sistem untuk memastikan user memasukan *password* yang benar tanpa menampung *password* dalam bentuk yang dapat digunakan untuk masuk.

Configuring Password Aging

Diagram berikut menghubungkan *relevant password aging parameter*, yang mana dapat disesuaikan dengan menggunakan *command* **change** untuk mengimplementasi *password aging policy*.




```
[user01@host ~]$ sudo chage -m 0 -M 90 -W 7 -I 14 user03
```

Command **change** diatas menggunakan opsi **-m, -M, -W** dan **-I** untuk memberikan *min days*, *max days*, *warn days*, dan *inactive days password user* masing-masing.

Command **change -d 0 user03** memaksa *user* **user03** untuk meng-*update password* pada *login* selanjutnya.

Command **change -l user03** menampilkan *detail aging password* **user03**.

Command **change -E 2019-08-05 user03** menyebabkan akun **user03** *expired* pada 2019-08-05 (YYYY-MM-DD)

Edit password aging configuration item di dalam *file* **/etc/login.defs** untuk mengatur *default password aging policies*. **PASS_MAX_DAYS** mengatur *default* maksimal umur dari *password*. **PASS_MIN_DAYS** mengatur *default* minimal umur *password*. **PASS_WARN_AGE** mengatur *default warning password*. Setiap perubahan di dalam *password default aging policies* akan efektif hanya untuk *user* baru saja. *User* yang sudah ada akan lanjut menggunakan *old password aging settings* daripada yang baru.

Restricting Access

Kita dapat menggunakan *command* **change** untuk mengatur tanggal akun kadaluarsa. Ketika tanggal tersebut tercapai, *user* tidak dapat *login* kedalam sistem secara interaktif. *Command* **usermod** dapat mengunci sebuah akun dengan *option* **-L**.

```
[user01@host ~]$ sudo usermod -L user03
[user01@host ~]$ su - user03
Password: redhat
su: Authentication failure
```

Jika *user* meninggalkan perusahaan, *administrator* mungkin mengunci dan mengakhiri akun dengan **usermod** *command*. Tanggal harus diberikan dengan format (YYYY-MM-DD).

```
[user01@host ~]$ sudo usermod -L -e 2019-10-05 user03
```

Command **usermod** diatas menggunakan opsi **-e** agar dapat mengatur **account expiry date** untuk diberikan kepada akun *user*. Opsi **-L** mengunci *password* milik *user*.

Mengunci akun mencegah *user* dari autentikasi ke dalam sistem menggunakan *password*. Ini merupakan metode yang direkomendasikan untuk mencegah akses ke sebuah akun oleh seorang pegawai yang telah *resign* dari perusahaan. Jika pegawai tersebut kembali, maka akun dapat dibuka kembali dengan **usermod -U**. Jika akun juga sudah *expired*, pastikan juga ubah *expiration date*.

The nologin Shell

nologin *shell* bertindak sebagai pengganti *shell* akun *user* yang tidak diharapkan untuk *login* ke dalam sistem secara interaktif. Hal ini merupakan sebuah kebijaksanaan dari sudut pandang *security* agar dapat melumpuhkan sebuah akun yang masuk kedalam sistem ketika akun tersebut tidak dibutuhkan. Contohnya, sebuah *mail server* mungkin membutuhkan sebuah akun untuk menyimpan *email* dan *password* user agar dapat mengautentikasi sebuah *mail client* yang digunakan sebagai pengambil *email*. Dapat disimpulkan *user* tersebut tidak membutuhkan akses *log directory* kedalam sistem.

Solusi yang masuk akal untuk situasi ini adalah untuk mengatur *login shell* milik *user* di **/sbin/nologin**. Jika *user* berupaya untuk *login* kedalam sistem secara langsung, **nologin** *shell* akan menutup koneksi.

```
[user01@host ~]$ usermod -s /sbin/nologin user03
[user01@host ~]$ su - user03
Last login: Wed Feb  6 17:03:06 IST 2019 on pts/0
This account is currently not available.
```

LEMBAR KERJA

KEGIATAN 1

Gaining Superuser Access

Pada kegiatan 1 kali ini akan mempraktikan *switching* **root** *account* dan menjalankan command sebagai **root**.

1. Siapkan *virtual machine* yang menggunakan OS CentOS dan masuk sebagai ***student***

2. Periksa *shell environment* milik ***student.*** Tampilkan *current user*, group information dan tampilkan *current* working directory. Juga tampilkan *environment variables* yang menunjukan *home directory* mirik *user* dan tunjukan lokasi *executable* milik *user*.
3. Switch ke **root** pada *non-login shell* dan explorasi *shell environment* yang baru (menampilkan detail apa saja seperti nomor 2).
4. Switch ke **root** pada *login shell* dan explorasi *shell environment* yang baru (menampilkan detail apa saja seperti nomor 2).
5. Periksa bahwa *user* **operator1** dikonfigurasi untuk menjalankan *command* apapun seperti *user* **sudo.**
6. Menjadi **operator1** dan buka *content* dari **/var/log/message**. Copy **/etc/motd** ke **/etc/motdOLD** dan hapus (**/etc/motdOLD**). Operasi ini membutuhkan hak *administrative*. Maka dari itu gunakan **sudo** untuk menjalankan *command* sebagai *superuser*. Jangan *switch* ke **root** menggunakan **sudo su** atau **sudo su -**. Gunakan **redhat** sebagai *password* **operator1**.

KEGIATAN 2

Managing Local User Accounts

Kegiatan 2 akan membuat beberapa *user* pada *system* kalian masing-masing dan *setting password* untuk *user* tersebut.

1. Siapkan *virtual machine* yang menggunakan OS CentOS dan masuk sebagai ***student***
2. *Switch* ke **root** menggunakan **sudo** pada *shell environment* user.
3. Buat *user* **operator1** dan pastikan bahwa **user** tersebut sudah terbuat di sistem.
4. *Set password* untuk **operator1** dengan *password* **redhat**.
5. Buat *user* tambahan bernama **operator2** dan **operator3**. *Set password* mereka menjadi **redhat**.
6. *Update* akun *user* **operator1** dan **operator2** agar masing-masing memuat *comment* Operator One dan Operator Two (operator1 = Operator One, operator2 = Operator Two). Pastikan *comment* berhasil ditambahkan.
7. Hapus *user* **operator3** beserta *personal data user* tersebut. Pastikan *user* berhasil dihapus sesuai dengan ketentuan.

KEGIATAN 3

Managing Local Group Accounts

Pada kegiatan 3, kalian akan membuat *group*, gunakan *group* tersebut sebagai *supplementary group* untuk beberapa *user* tanpa merubah *primary group user*, dan konfigurasi salah satu *group* dengan akses **sudo** agar dapat menjalankan *command* sebagai *root*.

1. Siapkan *virtual machine* yang menggunakan OS CentOS dan masuk sebagai ***student***
2. *Switch* ke **root** menggunakan **sudo**, dengan mewariskan *full environment user root*.
3. Buat *supplementary group* **operators** dengan GID 30000.
4. Buat **admin** sebagai *supplementary group* tambahan.
5. Periksa kedua *supplementary group* tersebut, dan pastikan apakah sudah terbuat atau belum.
6. Pastikan *user* **operator1**, **operator2** dan **operator3** tergabung dalam *group* **operators**.
7. Pastikan *user* **sysadmin1**, **sysadmin2** dan **sysadmin3** tergabung dalam *group* **admin**. Aktifkan hak *administrative* semua anggota *group* **admin**. Periksa semua member **admin** apakah dapat menjalankan administrative command atau tidak.

KEGIATAN 4

Kegiatan 4 kalian akan memberikan aturan terhadap *password* untuk beberapa *user*.

1. Siapkan *virtual machine* yang menggunakan OS CentOS dan masuk sebagai ***student***
2. Eksplorasi akun *user* yang ter-*lock* dan unlock sebagai **student**.
3. Ubah aturan *password* akun **operator1** agar memperbarui *password* setiap 90 hari. Pastikan umur *password* berhasil diatur.
4. Paksakan perubahan *password* pada *login* pertama untuk akun **operator1**.
5. *Login* sebagai **operator1** dan ubah *password* menjadi **forsooth123**. Setelah berhasil *setting password*, Kembali ke *user shell* **student**.
6. *Set* akun **operator1** agar memiliki kadaluarsa sejumlah 180 hari dari hari ini. *Hint*: **date -d "+180 days"** memberikan kalian tanggal dan waktu dari tanggal dan waktu saat ini.
7. *Set password* agar kadaluarsa 180 hari dari tanggal hari ini untuk semua *user*. Gunakan hak *administrative* untuk *edit file* konfigurasi.

RUBRIK PENILAIAN

Aspek Penilaian	Bobot Penilaian
Ketepatan menjawab semua kegiatan	65%
Pemahaman setiap aspek materi yang dibahas	20%
Quiz pertemuan materi minggu pertama	15%
Total	**100%**

*yok bisa yok, cheers for additional cheat sheet:")