

# Access Control List pada filesystem Linux

**Hak akses file Linux yang semula terbatas pada r,w, dan x harus diakui tidaklah selengkap sistem operasi lain. Untunglah, kini, kita dapat mempergunakan sistem hak akses berbasis *access control* yang lebih baik. Tutorial ini akan membahas bagaimana mengatur filesystem dengan kemampuan acl, serta membahas beberapa cara penggunaan hak akses tersebut.**

**B**icara soal keamanan server, tentunya tidak akan terlepas dari masalah keamanan filesystem. Ini adalah masalah yang mendasar. Pada sistem multiuser, adalah sangat tidak bijak apabila user yang satu dapat mengakses file milik user lain. Pemilik file sebenarnya dapat saja menutup semua akses dari luar ke file-file miliknya namun, akan kesulitan apabila beliau ingin membagi satu dua file ke pihak lain.

Di Linux, kita tidak mengenal istilah membagi file ke user atau group tertentu secara eksplisit. Yang kita miliki adalah hak akses sebagai pemilik, memberi akses kepada group pemilik file dan memberi akses kepada dunia luar. Untuk lebih jelasnya, berikut ini adalah gambaran hak akses sebuah file:

```
-rw-r--r-- 1 koljeng users 0
2005-02-10 09:59 test
```

Pada contoh tersebut, file test dimiliki oleh user koljeng, dan user tersebut memiliki hak baca dan tulis (dinotasikan dengan -rw-----). Sementara, anggota group users memiliki hak baca saja (dinotasikan dengan ---r-----) dan dunia luar juga memiliki hak baca (-----r--). Sebelum hadirnya fasilitas access control di Linux, akan sangat susah apabila user koljeng ingin membagi file hanya kepada satu user, misal user nop (dalam group users) namun tidak ke user lainnya (yang juga dalam group users). Hampir tidak mungkin dengan pendekatan hak akses filesystem apabila tetap menggunakan group users.

Kompromi yang sangat umum dilakukan adalah dengan membuat masing-masing

user dalam masing-masing group yang berbeda. Pendekatan ini, setahu penulis digunakan oleh distro Red Hat dan turunannya. Jadi, nama group seorang user adalah sama dengan nama user-nya. Apabila nama usernya koljeng, maka user koljeng ini juga menjadi anggota group koljeng. Dengan demikian, apabila suatu file dimiliki oleh user koljeng dan group koljeng, dan user tersebut ingin membagi file ke user nop (dan hanya kepada nop), maka koljeng dapat menggunakan beberapa cara. Salah satunya adalah dengan mengubah kepemilikan group (membutuhkan hak root), atau dengan menjadikan user nop anggota group koljeng (cara yang lebih baik, tetap membutuhkan root). Walaupun kedua cara tersebut membutuhkan hak root, tapi setidaknya, berbagi file ke user tertentu bisa dilakukan.

Pendekatan ini, bahwa seorang user memiliki group sendiri, menurut Anda, apakah merupakan pendekatan yang baik? Menurut penulis, rasanya bukan yang terbaik, walaupun dalam beberapa hal menyelesaikan solusi berbagi file kepada user tertentu (dengan beberapa kompromi). Tapi, harap diingat, bahwa cara seperti ini juga tetap memicu masalah baru, walaupun lebih ringan. Masalah tersebut adalah ketika kita ingin berbagi file kepada beberapa user yang tergabung dalam group tertentu. Apabila dengan gambaran pertama bahwa setiap user menjadi anggota group users, maka ini gampang sekali. Di sistem di mana setiap user memiliki group sendiri, untuk berbagi file kepada beberapa user yang tergabung dalam

group tertentu, kita bisa saja menggunakan dua cara berikut. Yang pertama, menjadikan beberapa user tersebut sebagai anggota group kita, atau membuat group baru yang beranggotakan beberapa user tadi. Tentunya, kepemilikan file harus disesuaikan. Di Linux, seorang user memang bisa menjadi anggota beberapa group.

Bagi pengguna rumahan, masalah berbagi file ini bukan masalah besar. Namun, penulis telah mendapati beberapa perusahaan yang tidak bisa berpindah hanya karena masalah hak akses ini. Wajar saja karena beberapa sistem operasi lain (hampir semua UNIX komersial, FreeBSD, Windows Server) memiliki fleksibilitas yang lebih besar dalam hal ini. Ini tentu saja juga termasuk Novell yang sejak versi-versi awalnya sangat memperhatikan masalah hak akses file.

Sebuah standar dengan nama IEEE POSIX 1003.1e draft 17, atau yang dikenal sebagai POSIX.1e dibuat untuk standarisasi ACL (Access Control List) untuk file dan direktori. Banyak sistem operasi yang kompatibel dengan standar POSIX telah mengadopsi dan mengimplementasikan standar ini. Termasuk Linux 2.6 yang telah mengimplementasikannya dengan sangat baik (di beberapa distro besar yang menggunakan 2.4, backport fasilitas ini mungkin tersedia). Di kernel 2.6, dukungan untuk ACL terdapat dalam hampir semua filesystem populer, yaitu EXT2, EXT3, XFS, ReiserFS, dan JFS, serta tidak tertutup kemungkinan untuk filesystem lain.

Sebenarnya, apakah ACL itu? Singkatan ini dalam masalah keamanan sistem cukup

populer. Dari namanya, Access Control, fungsinya adalah memungkinkan adanya akses control yang *fine-grained* terhadap *resource*. Jadi, untuk sistem yang mengimplementasikan ACL dengan sepenuhnya, berbagi resource sangatlah fleksibel untuk dilakukan. Di Windows, kita mengenal pula istilah ACE, yang menunjuk kepada entri yang diikutsertakan dalam ACL.

Bagi Anda yang menggunakan Linux sebagai server, atau ingin lebih baik lagi dalam mengelola keamanan filesystem dan lebih mudah dalam membagi resource filesystem, ACL sangat disarankan untuk digunakan.

## Kebutuhan sistem dan partisi

Kebutuhan sistem untuk dapat menggunakan ACL sangatlah sederhana. Apalagi, jika Anda telah menggunakan kernel 2.6. Yang perlu diperhatikan adalah pada saat melakukan *mounting* partisi dan paket *acl* untuk bekerja dengan ACL itu sendiri. Paket *acl* bisa Anda dapatkan di <http://acl.bestbits.at>, walau hampir semua distro Linux umum telah menyediakan paket tersebut dalam CDROM distro. Kalau pun tidak, Anda selalu bisa mencari paket untuk distro Anda di Internet.

Sebelum bekerja, untuk mudahnya, pastikan Anda telah menggunakan distro dengan kernel 2.6. Fedora Core 2 dan SUSE 9.1 adalah contoh distro dengan kernel 2.6. Kemudian, pastikan Anda tidak mencoba ACL di lingkungan kerja produktif. Sebaiknya Anda memiliki partisi kosong untuk coba-coba, atau Anda dapat mempergunakan partisi *home* Anda (apabila dibuat berbeda dengan partisi *root*). Sekali lagi, sebaiknya tidak di lingkungan kerja produktif.

Satu catatan, seperti Fedora Core 2 tidak memiliki dukungan untuk filesystem ReiserFS untuk di-mount dengan fasilitas ACL. Apabila menggunakan distro Fedora Core 2, Anda bisa mencoba dengan filesystem EXT2, EXT3 ataupun XFS. Artikel ini akan menggunakan distro SUSE, dengan filesystem ReiserFS, dan penulis akan bekerja pada partisi *root* (pada lingkungan non produktif, untuk coba-coba). Secara default, SUSE telah menambahkan fasilitas ACL tanpa Anda minta.

Amatilah *fstab* Anda. Berikut ini adalah contoh *fstab* penulis:

```
/dev/hda5 / reiserfs
acl,user_xattr 1 1
```

```
/dev/hda1 /data1 auto
noauto,user 0 0
/dev/hda2 /data2 auto
noauto,user 0 0
/dev/hda3 swap swap
pri=42 0 0
devpts /dev/pts devpts
mode=0620,gid=5 0 0
proc /proc proc
defaults 0 0
usbfs /proc/bus/usb usbfs
noauto 0 0
sysfs /sys sysfs
noauto 0 0
/dev/cdrom /media/cdrom subfs
fs=cdfss,ro,procuid,nosuid,nodev,
exec,icharset=utf8 0 0
/dev/fd0 /media/floppy subfs
fs=floppyfss,procuid,nodev,
nosuid,sync 0 0
```

Pada partisi untuk bekerja dengan ACL, dalam hal ini adalah */*, pastikan Anda memiliki opsi *mount acl* seperti pada baris berikut:

```
/dev/hda5 / reiserfs
acl,user_xattr 1 1
```

Anda bisa menggunakan program *mount* dengan opsi *(-o)* *acl* ketika ingin melakukan *mounting* tanpa melalui *fstab*.

Setelah Anda memiliki partisi yang di-mount dengan fasilitas ini dan dengan paket *acl* telah terinstal di sistem, maka kita telah siap bekerja dengan ACL.

## ACL dan utility lain

Ketika ACL diperkenalkan dan diimplementasikan, beberapa utility seperti *cp*, *mv*, dan *ls* telah dipersiapkan untuk mampu bekerja dengan ACL. Namun sayangnya, aplikasi untuk archive seperti *tar*, *cpio*, *pax* dan *dump* tidak akan menyimpan informasi tentang ACL. Apabila Anda tetap ingin mempergunakan archive, sangat disarankan untuk menggunakan program *star*. *Star* adalah program yang mirip dengan *tar*, namun menawarkan fasilitas ACL.

## Bekerja dengan acl: get dan set

Pada dasarnya, Anda cukup mempergunakan dua program untuk bekerja dengan *acl*, yaitu *getfacl* dan *setfacl*. Yang pertama untuk mendapatkan informasi ACL, yang

kedua untuk memberikan hak akses ACL. Mudah diingat, bukan?

Untuk mudahnya, kita akan melihat penggunaan *getfacl* terlebih dahulu. Penulis akan mempergunakan file *test* yang kita lihat pada awal tulisan. Berikut ini adalah contohnya:

```
$ ls -al test
-rw-r--r-- 1 koljeng users 0
2005-02-10 09:59 test

$ getfacl test
# file: test
# owner: koljeng
# group: users
user::rw-
group::r--
other::r--
```

Pada perintah *ls*, kita melihat bahwa file *test* dimiliki oleh user *koljeng*, group *users*, dan hak akses adalah bisa dibaca tulis oleh *koljeng*, bisa dibaca oleh group *users* dan dunia luar. Pada perintah *getfacl*, kita dapat melihat hak akses ini apa adanya, tanpa fasilitas ACL.

Mulai sekarang, file *test* ini akan kita buat hanya bisa dibaca tulis oleh user *koljeng*, sehingga pada *ls* akan terlihat seperti berikut:

```
-rw----- 1 koljeng users 0
2005-02-10 09:59 test

$ getfacl test
# file: test
# owner: koljeng
# group: users
user::rw-
group::---
other::---
```

Apabila user *nop* ingin mengakses file *test* tersebut, maka pesan kesalahan seperti ini akan ditampilkan:

```
$ cat test
cat: test: Permission denied
```

Setelah melihat penggunaan *getfacl*, kita akan melihat penggunaan *setfacl*. Contoh yang pertama adalah mengizinkan user *nop* untuk membaca file tersebut. Sebagai informasi tambahan, user *nop* dan *koljeng* sama-sama termasuk dalam group *users*. User lain yang tergabung dalam group *user*, yaitu

buskota, tidak diizinkan untuk mengakses file test tersebut.

```
$ setfacl -m u:nop:r-- test
```

Dengan perintah tersebut, kini user nop telah dapat membaca file tersebut. Kita akan melihat bagaimana tampilan ls dan getfacl terhadap file ini:

```
$ ls -al test
-rw-r-----+ 1 koljeng users 0
2005-02-10 09:59 test
```

```
$ getfacl test
# file: test
# owner: koljeng
# group: users
user::rw-
user:nop:r--
group::---
mask::r--
other::---
```

Perhatikanlah tampilan keluaran ls. Ada yang berubah. Yang pertama adalah munculnya hak akses r pada group (padahal kita tidak melakukannya). Yang kedua adalah munculnya tanda + setelah hak akses. Mengenai munculnya r pada hak akses group, ini adalah notasi semata. User buskota, yang tidak memiliki hak baca padahal tergabung dalam group users, tetap tidak dapat membaca file ini. Bukti:

```
$ id
uid=1002(buskota) gid=100(users)
groups=14(uucp),16(dialout),17(audio),33(video),100(users)
# cat test
cat: test: Permission denied
```

Jadi, sekali lagi, r tersebut hanyalah sebagai penanda untuk penggunaan ACL.

Program setfacl dapat pula digunakan untuk menggantikan program chmod. Caranya, sama seperti sebelumnya, hanya kita tidak menyebutkan username di opsi -m. Apabila kita menggunakan -m u:nop:r--, maka untuk menggantikan chmod, kita akan menggunakan -m u::rw-,g::---,o::---, sebagai contoh:

```
$ setfacl -m o::rw-,g::---,
o::--- test
```

Dengan cara yang sama, Anda bisa memberikan hak untuk group lain atau user lain

mempergunakan perintah setfacl ini.

Setelah memberikan akses, kita dapat pula menghapus akses user nop pada file test tersebut dengan opsi -x, seperti contoh berikut:

```
$ setfacl -x u:nop test
```

Kini, user nop tidak lagi dapat membaca file test tersebut. Berikut ini adalah tampilan dari ls dan getfacl:

```
$ ls -al test
-rw-----+ 1 koljeng users 0
2005-02-10 09:59 test
```

```
$ getfacl test
# file: test
# owner: koljeng
# group: users
user::rw-
group::---
mask::---
other::---
```

Walaupun kita menghapus ACL untuk user nop, file tersebut tetap masih memiliki atribut ACL walaupun tidak ada hak yang diberikan kepada user lain (karena ACL masih berlaku pada pemilik file, tentunya). Untuk menghapus ACL secara keseluruhan pada file tersebut, gunakan opsi --remove-all seperti contoh berikut:

```
$ setfacl --remove-all test
```

Tampilan perintah ls pun kembali normal:

```
$ ls -al test
-rw----- 1 koljeng users 0
2005-02-10 09:59 test
```

Bagi Anda yang ingin bekerja dengan ACL secara rekursif, berikanlah opsi -R seperti pada ls, chmod dan chown.

Pembahasan berikut adalah bagaimana mempergunakan file ACL untuk memberikan hak pada file tertentu. Sebagai contoh, berikut ini adalah isi file dengan nama acl.entry:

```
u:nop:r--
```

Contoh ini hanya berisikan satu baris ACL. Anda selalu bisa menambahkan baris lain, dengan mengulangi u, g atau o. Apabila Anda lebih suka mempergunakan bentuk panjang, Anda bisa mempergunakan user,

group dan other. Untuk bekerja dengan file tersebut, berikan opsi -M pada setfacl:

```
$ setfacl -M acl.entry test
```

Selain opsi -m yang kita selalu gunakan untuk memberikan ACL, kita mengenal pula opsi --set. Perbedaan dengan opsi -m adalah opsi --set akan mengatur ulang ACL, sementara opsi -m adalah untuk memodifikasi ACL. Anda bisa mempergunakan salah satu atau keduanya, tergantung situasi. Apabila ingin memodifikasi, gunakan -m, namun, apabila malas memodifikasi dan ingin mengatur ulang saja, gunakan --set.

## Default ACL


Kita telah bekerja dengan access ACL dan melihat bagaimana acl dapat dipergunakan untuk berbagi file. Sebenarnya, selain access ACL, ada pula istilah default ACL. Default ACL adalah ACL yang dikenakan pada direktori saja, dan sebagai akibat dari penggunaan default ACL ini, file dan direktori yang dibuat di dalam direktori yang dikenakan default ACL tersebut akan menurunkan dari direktori tersebut.

Untuk mengenakan default ACL pada suatu direktori, caranya sama saja dengan cara sebelumnya, namun kita akan menambahkan opsi -d. Contoh:

```
$ setfacl -d --set u::rw,u:nop:
r--,g::---,o::--- dir1/
```

Berikut ini adalah keluaran getfacl dir1:

```
$ getfacl dir1/
# file: dir1
# owner: koljeng
# group: users
user::rw-
group::---
other::---
default:user::rw-
default:user:nop:r--
default:group::---
default:mask::r--
default:other::---
```

Dengan ACL, kita bisa mengatur access control tanpa harus menyibukkan sysadmin. Ini tentu berbeda sekali dengan ilustrasi pada awal artikel ketika harus membuat group baru, menambahkan user ke suatu group dan lain sebagainya. 

**Noprianto** ([noprianto@infolinux.co.id](mailto:noprianto@infolinux.co.id))