Filsikkerhet i Linux

Filer og brukere

- Standard Linux tilbyr filsikkerhet på *brukernivå*
- Alle brukere tilhører en eller flere *grupper* av brukere, med ett eller flere gruppemedlemmer
- Alle filer har tilhørighet til en *eier*, som er en av de registrerte brukerne på systemet
- Alle *filer* har også tilhørighet til *en* gruppe av brukere

Filtilgang for ulike brukere

Tilgangsrettighetene til en fil settes *separat* for:

- **u** eier (**u**ser)
- **g** gruppe (**g**roup)
- **o** alle andre brukere (**o**thers, “rest of the world”)

Rettigheter til filer

En bruker kan ha tre rettigheter til en fil:

`r` Lesetilgang – **r**ead permission

Filen kan åpnes (read-only hvis ikke skrivetilgang) og vises med kommandoer som `cat` og `more`

`w` Skrivetilgang – **w**rite permission

Innholdet av filen kan endres, filen kan overskrives

`x` Kjøretilgang – e**x**ecute permission

Filen kan eksekveres (kjøres) hvis den er et eksverbart program (ferdig kompilert maskinkode) eller et script som kan utføres av en interpreter

Tilgang og rettigheter til kataloger

For kataloger (som også er filer) betyr rettighetene:

`r` Lesetilgang / read permission

Innholdet i katalogen kan listes (med `ls`)

`w` Skrivetilgang / write permission

Kan opprette, flytte og fjerne filer i katalogen

`x` Kjøretilgang / execute permission

Kan gjøre en `cd` til katalogen

`ls -l` : Se tilgangsrettighetene til en fil

```
ls -l a.txt
-rwxr-xr-- 1 janh janh 399 Apr 12  2013 a.txt
```

- Første tegn angir filtypen (`-` betyr “regulær fil”)
- De ni neste tegnene er gruppert tre og tre for **`u`**, **`o`** og **`g`** :

`rwx` Eier kan lese, skrive og kjøre filen (alle rettigheter)

`r-x` Gruppen kan lese og kjøre filen, men ikke endre

`r--` Andre kan bare lese filen, men ikke endre og kjøre

`chmod` – Endre tilgangsrettighetene til en fil

`chmod` ***`tilgang filnavn`***

- Kan angi *endringer* i tilgang med `+` og `-` :

  `u+w` Legger til skrivetilgang for eier

  `o-r` Fjerner lesetilgang for “others”

  `a+r` Gir lesetilgang til alle brukere (`a` = all)

- Kan også angi tilgang eksplisitt med `=` :

  `u=rwx` Gir alle rettigheter til eier

  `g=rx` Gruppen får lese- og skrive-, ikke kjøretilgang

  `o=` “Others” får ingen tilgang

chmod – Eksempler

```
$ ls -l a.cpp
-rwxr-xr-- 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod o+x a.cpp
$ ls -l a.cpp
-rwxr-xr-x 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod u-w a.cpp
$ ls -l a.cpp
-r-xr-xr-x 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod oug+w a.cpp
$ ls -l a.cpp
-rwxrwxrwx 1 janh janh 0 Sep 12 14:56 a.cpp
```

Representasjon av tilgangsrettigheter

- Tilgangsrettigheter lagres som 9 bits (0 eller 1)
- 1 betyr at man har en rettighet, 0 ingen rettighet
- Tre bits representerer `rwx` for hver brukergruppe
- Rettighetene for *en* brukertype (u, g, o) kan derfor tolkes som et *oktalt* tall (i åttetallsystemet)
- Tilgangen til en fil for *alle* tre brukertyper kan representeres med tre oktale siffer (0,1,2,…,7)
- Kompakt skrivemåte som kalles for en “file mode”

File mode

- Tresifret oktalt tall som lagrer tilgangsrettigheter:

1. siffer	eier av filen	**`u`**
2. siffer	filens gruppe	**`g`**
3. siffer	andre brukere	**`o`**

- Hver rettighet har en unik verdi:

4	read	**`r`**
2	write	**`w`**
1	execute	**`x`**
0	ikke rettighet	**`-`**

Beregning av file mode

- *Legger sammen* rettighetene for hver brukergruppe
- Får et *unikt* tall for hver mulig kombinasjon rettigheter:

7 = 4+2+1	`rwx`	**3** = 0+2+1	`-wx`
6 = 4+2+0	`rw-`	**2** = 0+2+0	`-w-`
5 = 4+0+1	`r-x`	**1** = 0+1+0	`--x`
4 = 4+0+0	`r--`	**0** = 0+0+0	`---`

Bruk av `chmod` med file mode

```
$ chmod 644 a.cpp
$ ls -l a.cpp
-rw-r--r-- 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod 700 a.cpp
$ ls -l a.cpp
-rwx------ 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod 521 a.cpp
$ ls -l a.cpp
-r-x-w---x 1 janh janh 0 Sep 12 14:56 a.cpp
$ chmod 400 a.cpp
$ ls -l a.cpp
-r-------- 1 janh janh 0 Sep 12 14:56 a.cpp
```

Noen eksempler på file modes (1)

400	**r--------**	Lesetilgang bare for bruker. For “hemmelige” dokumenter(?)
644	**rw-r--r--**	Lese/skrive for bruker, andre lese. Mye brukt ved fildeling og for websider.
664	**rw-rw-r--**	Bruker og gruppe lese/skrive, andre lese. Vanlig ved fildeling i arbeidsgrupper.
755	**rwxr-xr-x**	Legger til kjøretilgang på 644 for alle. Vanlig for delte kataloger / web.

Noen eksempler på file modes (2)

745	**rwxr--r-x**	Kjøretilgang for eier og “verden”. Alternativ til 755.
711	**rwxr--r-x**	Gruppe og “verden” kan kjøre. Vanlig for kataloger på webservere.
666	**rw-rw-rw-**	Farlig (number of the beast)
777	**rwxrwxrwx**	Jeg er idiot

GUI og filbeskyttelse

- Filtilgang kan også sette med pek-og-klikk
- Bruk en file browser:
  - Klikk på “permissions”
  - Sett tilgang individuelt for eier, gruppe og bruker
- Tungvint, langsomt
- Ubrukbart for håndtering av mange filer:
  - `find` kombinert med `chmod` er løsningen

Avansert kontroll av filtilgang

- SELinux:
  - Security Enhanced Linux tilbyr utvidete beskyttelsesmekanismer (kapittel 8 i læreboken)
- Utvidet bruk av eksekveringsrettighetene for bruker og gruppe
  - Kan kontrollere bruker-ID og gruppe-ID til et kjørende program (kapittel 4 i læreboken)
- Sticky bit
  - Brukes spesielt til å kontrollere tilgang til en katalog der “alle” har skriverettigheter

Sticky Bit *

- Brukes for ekstra kontroll på helt åpne kataloger
- Typisk en katalog som `/tmp`, der vi tillater at “alle” kan legge fra seg filer (f.eks. deling av bilder)
- Vi kan legge til et “sticky bit” for å hindre brukere i å slette eller endre filer som de ikke eier selv
- “When the sticky bit is set on a directory, files in that directory may only be removed or renamed by root or the directory owner or the file owner”

*: Betegnelsen “sticky” for slike kontrollbits er over 40 år gammel, men brukes fortsatt, selv om anvendelsen nå er helt anderledes enn i eldre OS/Unix

Sette og fjerne sticky bit

- Sette sticky bit på katalog der alle har tilgang:

  `chmod 1777 katalog`

  `chmod o+t katalog`

- Fjerne sticky bit:

  `chmod 777 katalog`

  `chmod o-t katalog`

- Liste katalog med sticky bit satt, med `ls -ld`:

```
drwxrwxrwt 3 janh janh 4096 sep.1 11:51 katalog
```