

Linux filesystem permissions

David Morgan

© David Morgan 2003,2004

An access control mechanism

- For granting/withholding access to a resource
- Based on relation between file- and user-characteristics
- Analogy
 - government documents receive classifications
 - government employees receive clearances
 - access to particular document by particular employee determined by relation between classification and clearance

© David Morgan 2003,2004

ls -l shows permissions

```
[root@sputnik joe]# ls -l      permission set
total 52
drwxr-xr-x 4 joe joe 4096 Mar 13 06:05 Desktop/
-rw-r--r-- 1 david sales 524 Apr 8 21:30 adbudget
-rw-r--r-- 1 root root 30720 Apr 8 21:12 classifications.tar
-rw-r--r-- 1 joe sales 1620 Apr 8 21:29 mycalendar
-rw----- 1 joe joe 3302 Apr 8 21:29 mypaystubs
drwx----- 2 joe joe 4096 Mar 13 06:05 tmp/
[root@sputnik joe]#
```

© David Morgan 2003,2004

ls -l shows a user & group associated with each file

```
[root@sputnik joe]# ls -l      user associated with file adbudget
total 52
drwxr-xr-x 4 joe joe 4096 Mar 13 06:05 Desktop/
-rw-r--r-- 1 david sales 524 Apr 8 21:30 adbudget
-rw-r--r-- 1 root root 30720 Apr 8 21:12 classifications.tar
-rw-r--r-- 1 joe sales 1620 Apr 8 21:29 mycalendar
-rw----- 1 joe joe 3302 Apr 8 21:29 mypaystubs
drwx----- 2 joe joe 4096 Mar 13 06:05 tmp/
[root@sputnik joe]#      group associated with file mycalendar
```

© David Morgan 2003,2004

File system - permissions

-rwxr-x---

- **File type** (file, directory, device,...)
- Accesses granted to **file's associated User**
- Accesses granted to members of **file's Group***
- Accesses granted to all **Other users**

*other than the associated user

© David Morgan 2003,2004

Meaning for files

- **r** – read
 - can open file
- **w** – write
 - can modify, overwrite, truncate
- **x** – execute
 - can execute (in manner indicated in 1st two bytes)

© David Morgan 2003,2004

Meaning for directories

- **r** – read
 - can view contained files
- **w** – write
 - can change contained files (add, rename, remove)
- **x** – execute
 - can enter directory (cd)
 - can open contained files in directory or its subs

© David Morgan 2003,2004

chmod – change file permissions

- To restrict/extend access to others
- To enable script execution

© David Morgan 2003,2004

chmod – change granularity

- entire
 - use octal specification
- surgical
 - use who/how/what specification

© David Morgan 2003,2004

changing all permissions – octal specification

- - -	0 0 0	0	Used in triples: e.g., 750 = rwXr-x---
- - X	0 0 1	1	
- W -	0 1 0	2	
- W X	0 1 1	3	
r - -	1 0 0	4	
r - X	1 0 1	5	
r W -	1 1 0	6	
r W X	1 1 1	7	

© David Morgan 2003,2004

changing just some permissions – who/how/what specification

who	how	what
u	+	r
g	-	w
o	=	x
a		s

© David Morgan 2003,2004

who/how/what

- **u** – for that user associated with the file (“owner”)
- **g** – for those users in group associated with the file
- **o** – for anybody else (“world”)
- **a** – all three of them

© David Morgan 2003,2004

who/how/what

- + add, other existing permissions unaffected
- - remove, other existing permissions unaffected
- = set, existing permissions replaced

© David Morgan 2003,2004

who/how/what

- r - read
- w - write
- x - execute
- s - establish “set id” behavior

© David Morgan 2003,2004

Processes and users

- Running processes are associated with user(s)
 - real user/UID –id of user running process
 - effective user/UID – id of user owning executable
- process's real and effective UIDs are same, usually

© David Morgan 2003,2004

SUID – exception to the rule

- SUID – a permission characteristic of files
- changes the effective UID of file's process when run
 - from UID of user who runs the program
 - to UID of user who “owns” the file

© David Morgan 2003,2004

SUID – exception to the rule

- WHAT – “When a SUID file is run, the process involved takes on an *effective UID* that is the same as the owner of the file.”
- WHY – “Sometimes, unprivileged users must be able to accomplish tasks that require privileges.
- EXAMPLES – passwd, mail

© David Morgan 2003,2004

SUID shell scripts

- BAD
- DON'T
- Security flaw – launches SUID shell to run script

© David Morgan 2003,2004

sudo – secure solution

- lets certain user(s) run certain program(s) as another user
- user runs program indirectly under sudo's control: `sudo <targetprogram>`
- sudo configuration defines who can run what as whom

© David Morgan 2003,2004

Some permissions

```
ls -ld /
drwxr-xr-x    root    root    /

/home:
drwxr-xr-x    root    root    /home

/home/joe:
drwx-----   joe     joe     /home/joe

/home/david:
drwx-----   david   david   /home/david

/root
drwxr-x---    root    root    /root
```

© David Morgan 2003,2004

