Authorization in UNIX

System Administration Decal
Spring 2008
Lecture #3
Joshua Kwan

Last time…

- Filesystem hierarchy in UNIX: /dev, /usr, /bin, /sbin, etc.
- Regular files, directories, links, device files, named pipes
- The difference hard links and symbolic links
- Why file extensions don't matter … that much

Homework Remarks

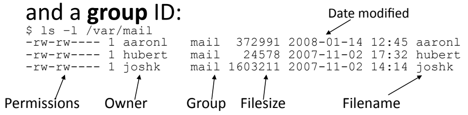
- Device nodes have **major** and **minor** numbers that designate the device class, and then the device number
- /proc/self is like a **dynamic** symlink!
- Mounts: If it ain't in /dev, it's either
  - – A 'fake' filesystem (/proc, ramdisks)
  - – A bind mount
  - – Network share
  - – Loopback mount (i.e. mounting an ISO file)

Today

- Review of basic permissions
- When you log in to a UNIX machine, how is your password checked?
- Special types of permissions: sticky bit, setgid, setuid
- More fine grained permission control: ACLs

Owners, Groups, Permissions

- In UNIX, every file associated with a **user** ID and a **group** ID:



```
$ ls -l /var/mail
-rw-rw---- 1 aaronl   mail  372991 2008-01-14 12:45 aaronl
-rw-rw---- 1 hubert   mail   24578 2007-11-02 17:32 hubert
-rw-rw---- 1 joshk    mail 1603211 2007-11-02 14:14 joshk
```

- Many users can be in a single group; one user can be in many groups.
- Here, `aaronl` can read/write his mail file, and members of group `mail` also can

Owners, Groups, Permissions

- Utilities that help you do this stuff!
  - – **chmod** – Change the permissions on a file.
  - – **chown** – Change the owner of a file.
  - – **chgrp** – Change the group association of a file.
- Remember, you can set permissions individually for each set of users: the **owner**, **group**, or **everyone else**.

Owners, Groups, Permissions

- 3 types of file permissions:
  - Read: the ability to read the content of the file.
  - Write: the ability to modify the file.
  - Execute: the file can be run as a program.
- New permissions:
  - Sticky bit: All files created in dir. will have GID of dir.
  - Setuid: Executables run as user who owns the file (setuid root: anyone can run this, and run as root.)
  - Setgid: Executables run as group associated with file. (setgid games: useful for saving high scores, why?)

User and Group Information

- How is all of this data stored?
- Three files…
  - /etc/passwd: Stores user name, user ID, and personal information. (World readable)
  - /etc/shadow: Contains mapping from user name to password (Only readable/writable by root)
  - /etc/group: Contains group names, group IDs, and members of the group (World readable)
- Use **getent** tool to look things up in these files

Examples

- A passwd entry

```
joshk:x:1000:1000:Joshua Kwan,208,,:/home/joshk:/bin/zsh
```

username uid primary gid my name room # etc. home directory login shell

- A shadow entry

```
joshk:$1$/SWUUnPr$lx2ILBmkfsd61dTbMi.lQ.:13037:0:99999:7:::
```

username encrypted password (MD5 hash) lots of "days since…"

- A group entry

```
wheel:x:500:wjm,joshk
```

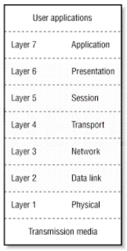
name password gid members

Diversion (Cool Stuff)

- Many UNIX systems use the passwd/group/shadow method of authentication
- NIS: Network Information Service
- LDAP: Lightweight Directory Access Protocol
- Using a system called PAM (Pluggable Authentication Modules), you can use *anything* for authentication!

Diversion (Cool Stuff)

- Remember the seven-layer OSI model
- Intent: Each layer uses layer below it to provide service to layer above it
- If you replace a low layer, you get all the rest for free!



User applications

Layer 7	Application
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data link
Layer 1	Physical

Transmission media

IP over Avian Carrier

- RFC 1149 – look it up! (When was it written?)
- Carrier pigeons replace layer 1 – physical medium
- Some crazy Norwegians tried it out in 2001!

IP over Avian Carrier

- Results of the experiment:

```
vegard@gyversalen:~$ ping -i 900 10.0.3.1
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms
--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
```

Administrivia

- Please send me your homework in text format!
- This week, there *is* a homework.
- Lab will be posted by Thursday afternoon
- Start getting groups together for project
- What do you think about the labs? Too hard? Too easy? Too boring?

Access Control Lists

- Access Control Lists (ACLs) provide fine grained permissions
- "Iris can read or write this file, but Calvin can't do anything to it at all!"
- **setfacl**, **getfacl**: Set/get ACL for a particular file
- Most simple permission cases don't need ACLs, can be done with groups instead.

"sudo" as an Access Control List

- **sudo**: tool for letting normal users run certain things as root
- Like an ACL for privileged commands
- Managed with the "visudo" command

```
# User privilege specification
root    ALL=(ALL) ALL     (root may use all commands – duh)
%wheel  ALL=(ALL) ALL     (all in group wheel also may do everything)
wm      ALL=/usr/sbin/apache2ctl   (wm may only use apache2ctl)
```