

CSE 265:

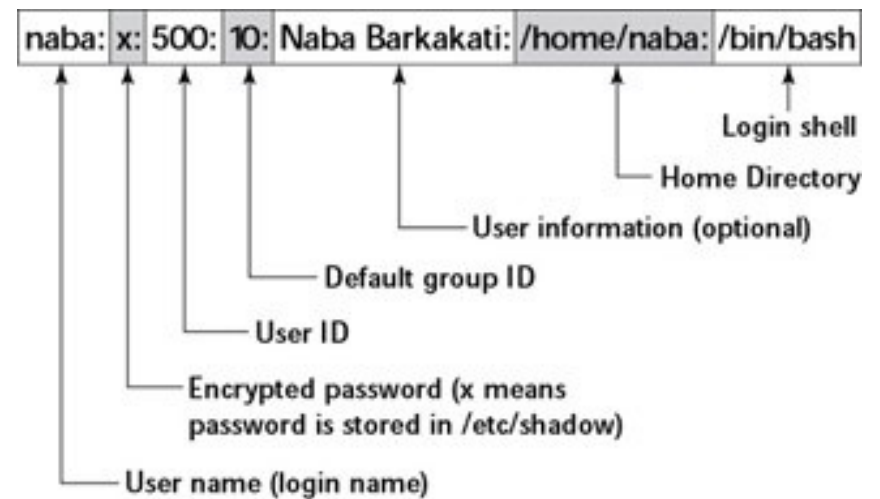
System and Network Administration

- User accounts
 - The /etc/passwd file
 - The /etc/shadow file
 - The /etc/group file
 - Adding users
 - Removing users
 - Disabling logins
 - Account management utilities
- Root powers
 - Ownership of files and processes
 - The superuser
 - Choosing a root password
 - Becoming root
 - Other pseudo-users

The /etc/passwd file

- /etc/passwd lists all recognized users and contains:

- login name
- encrypted password (unless /etc/shadow used)
- UID number
- default GID number
- full name, office, extension, home phone (optional)
- home directory
- login shell



- Examples

```
root:lga4FjuGpZ2so:0:0:The System,,x6096,:/:/bin/csh
jl:x:100:0:Jim Lane,ECT8-3,,:/staff/fl:/bin/sh
```

Login name

- Syntax
 - usernames must be unique
 - ≤ 32 chars
(old systems/NIS: limit 8 chars)
 - any characters except newlines and colons
- Recommendations
 - use lower case (even though case sensitive)
 - choose easy to remember
 - avoid “handles” and cutesy nicknames



Encrypted passwords

- Most passwords are in /etc/shadow, not /etc/passwd
- Passwords are stored encrypted
 - Cannot be changed by hand
 - Can be copied from another account
 - Are set using **passwd** (or **yppasswd** for NIS)
- Passwords should never be left blank
 - Put a star (*) in place (x for shadow usage)
 - Otherwise no pw needed!
- MD5 passwords (most distributions) can be any length
 - Other systems only use the first eight characters



UID number

- In Linux, UIDs are unsigned 32-bit integers (4B!)
 - Older systems only allowed up to 32,767
- Root is (almost always) UID 0
- Fake/system logins typically have low UIDs
 - Place real users ≥ 100
- Avoid recycling UIDs
 - Old files, backups are identified by UID
- Preserve unique UIDs across org
 - helpful for NFS

Other fields

- default GID number
 - like UIDs, 32-bit unsigned integers
 - GID – is for the group “root”
- GECOS fields (optional) [**chfn**]
 - General Electric Comprehensive OS
 - full name, office, extension, home phone
- home directory
 - Where the user starts when the log in
- login shell [**chsh**]
 - such as sh/bash, csh/tcsh, ksh, etc.



The /etc/shadow file



-
- Readable only by superuser
 - Enhanced account information
 - Use is highly recommended
 - Use **usermod** to modify contents
 - Contains:
 - Login name
 - Encrypted password
 - Date of pw change
 - Min number of days between password changes
 - Max days between pw changes
 - Num days in advance to warn
 - Num days after expiration to disable account
 - Account expiration date
 - Reserved field

The /etc/group file

- Contains names of groups and lists each member
- Example:
 - `wheel:*:10:root,evi,garth,trent,brian`
 - Group name:encrypted password:GID:List of members, separated by commas (no spaces)
- Setting per-user groups is recommended
 - Better default security

Adding users

- For small installations, adding users is simple
 - Have user sign and date user agreement
 - Create user account with `useradd`
 - Set password with `passwd`
 - Change defaults with `usermod`



Steps to add a user (1)

- Edit the `/etc/passwd` and `/etc/shadow` files to define account
 - Use `vipw` to lock and edit with `$EDITOR`
- Set an initial password
 - # passwd user**
- Create, `chown`, and `chmod` the user's home directory
 - # mkdir /home/staff/tyler**
 - # chown tyler.staff /home/staff/tyler**
 - # chmod 700 /home/staff/tyler**



Steps to add a user (2)

- Copy default startup files to the user's home directory
 - bash
 - .bashrc, .bash_profile
 - csh/tcsh
 - .login, .cshrc, .logout
 - X-windows
 - .Xdefaults, .Xclients, .xsession
- Need to create and store default files!



Steps to add a user (3)

- Copy files to new directory
 - # **cp /etc/skel/.[a-zA-Z]* ~tyler**
 - # **chmod 644 ~tyler/.[a-zA-Z]***
 - # **chown tyler ~tyler/.[a-zA-Z]***
 - # **chgrp staff ~tyler/.[a-zA-Z]***
- Cannot use **chown tyler ~tyler/.***
- Set mail home
 - might edit /etc/mail/aliases



Steps to add a user (4)

- Edit `/etc/group` file
 - Add to relevant groups
- Set disk quotas with `edquota`
- Verify new login
 - log in as new user
 - execute `pwd` and `ls -la`
- Notify new user of account and initial password
 - get signed AUP
- Record account status and contact information



Removing users

- Generally with **userdel**
 - Set disk quota to zero
 - Remove user from local databases or phone lists
 - Remove from aliases file (or add forwarding)
 - Remove crontab and any pending at jobs
 - Kill any running processes
 - Remove temporary files in /var/tmp or /tmp
 - Remove from passwd, shadow, and group files
 - Remove home directory (backup first) and mail spool

Disabling logins

- Sometimes need to temporarily disable a login
- Can't just put a star in front of encrypted pw
 - Might still be able to log in via network w/out pw
- Current practice
 - Replace shell with program explaining status and instructions on how to fix

Account management utilities

- Basic utilities
 - **useradd** – adds to passwd and shadow files
 - **usermod** – changes existing passwd entry
 - **userdel** – remove user, opt. delete home dir
 - **groupadd**, **groupmod**, **groupdel** operate on /etc/group
- Common to write custom **adduser** and **rmuser** scripts

The superuser



- The root account has UID of 0
 - Can change the name and create other users with same UID; neither recommended
- The superuser (any process with effective UID 0) can perform any valid operation on any file or process.
- All other users are “normal”

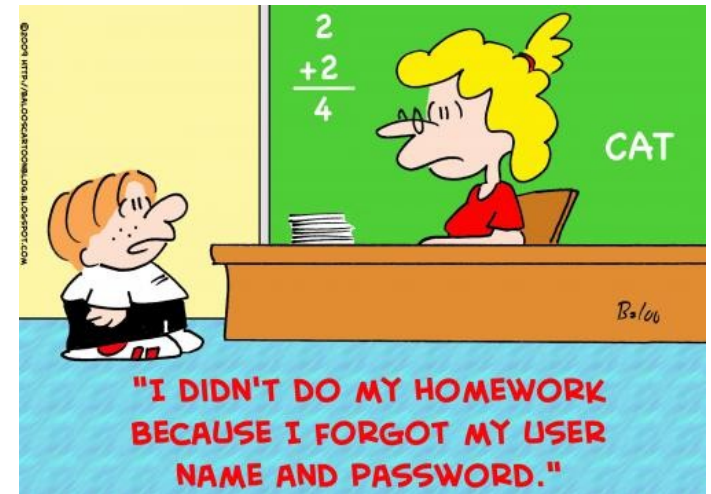
Restricted operations



- Superuser privileges are required for:
 - Changing the root directory of a process with chroot
 - Creating device files
 - Setting the system clock
 - Raising resource usage limits and process priorities
 - Setting the system's hostname
 - Configuring the network interfaces
 - Opening privileged network ports (≤ 1024)
 - Shutting down the system
 - Changing process UID and GID (only one way)
 - Example: login

Choosing a root password

- Any password? Not if you want it to be difficult to crack.
- Should be
 - At least eight characters (more may not be helpful)
 - Not easily guessed or found by trial and error
 - Memorable (so you don't need to write it down)
 - A seemingly random sequence of letters, digits, & punctuation
 - **Shocking nonsense!**
 - Memorable, unguessable, unique, undisclosed
 - Mpmgg!: "Mollusks peck my galloping genitals!"



Changing the root password

- Should be performed
 - At least every three months
 - Every time someone who might know the password leaves the site
 - Whenever you think security might be compromised
 - On a day when you will remember the new pw!



Change all the Passwords



Becoming root

- You can log in as root
 - No record of what operations were performed
 - Often you'll want a record!
 - When the root user was a colleague who is unavailable
 - When you can't remember exactly what you did
 - When the access was unauthorized and you want to know what was done
 - No record of who was root
- Typically want to disable root logins except at console



got root?

Being root

- Responsibilities!
 - Do not give out root password
 - Do not create new accounts with UID 0
 - Use root account for admin work only
 - Change root password often
 - Do not leave root shell unattended
 - Be extra careful!
 - Perhaps more, depending on policies at location



got root?

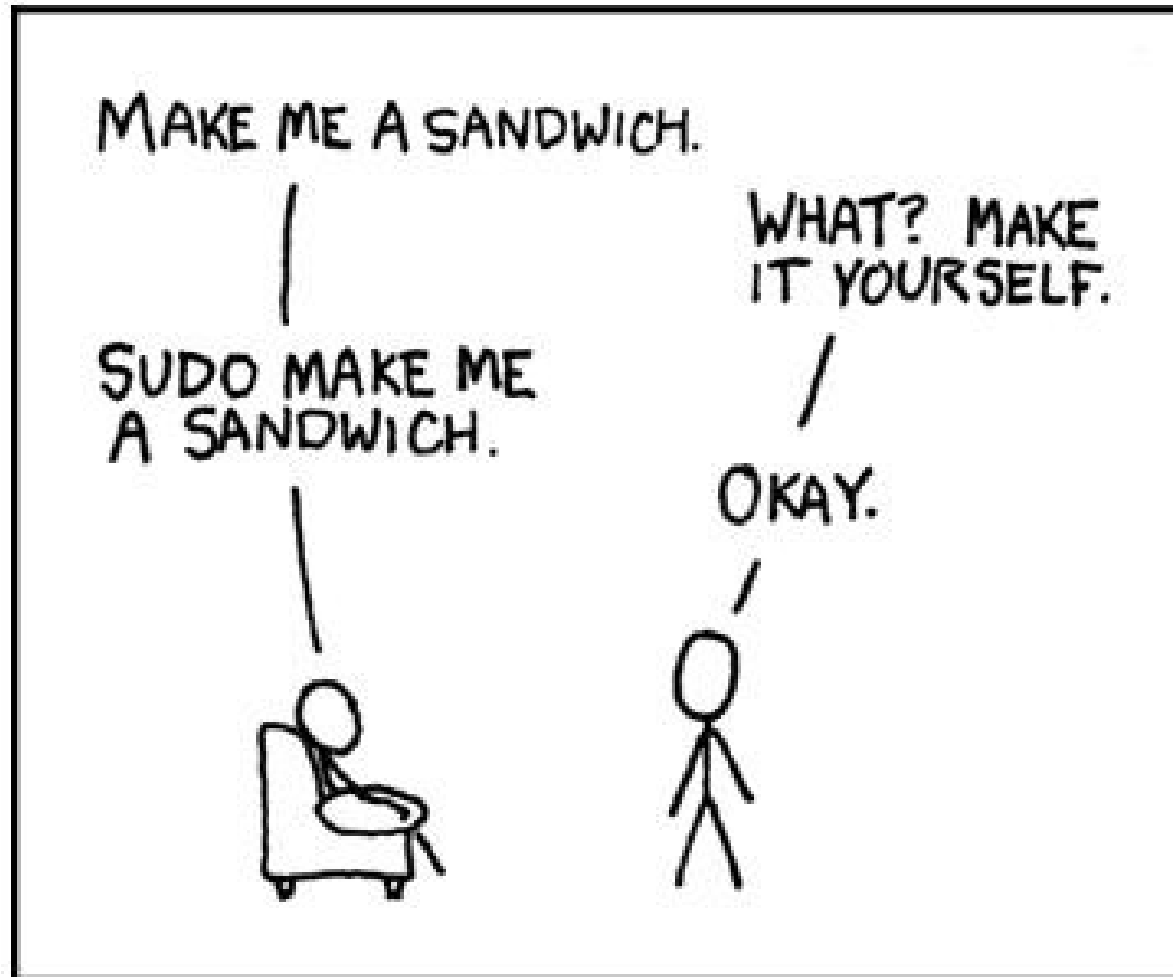
SU

- **su**: substitute user identity (switch users)
 - Without args, su prompts for root password and then starts root shell
 - Logs who became root and when
 - Can also **su *username***
 - if you know the pw, or are root already
 - Use “**su -**” to execute new user's shell
 - Otherwise new PATH is not established
 - Good idea to use full pathname to **su** (why?)
 - Linux: /bin/su
 - Solaris: /sbin/su

sudo

- sudo: a limited su
 - When you want to provide limited root-privileges
 - **sudo <program to be executed>**
 - Checks /etc/sudoers for authorization
 - Asks for user's password
 - Logs command executed, person, time, and directory
 - Executes command
 - Additional sudo commands can be executed without password for another five minutes
 - Example:
 - sudo /bin/cat /etc/sudoers

Famous XKCD



Example sudoers file

```
# Define aliases for machines in CS & Physics departments
```

```
Host_Alias    CS = tigger, anchor, piper, moet, sigi
```

```
Host_Alias    PHYSICS = eprince, pprince, icarus
```

```
# Define collections of commands
```

```
Cmnd_Alias    DUMP = /sbin/dump, /sbin/restore
```

```
Cmnd_Alias    PRINTING = /usr/sbin/lpc, /usr/bin/lprm
```

```
Cmnd_Alias    SHELLS = /bin/sh, /bin/csh/, /bin/bash, /bin/ash
```

```
#Permissions
```

```
mark, ed     PHYSICS = ALL
```

```
herb         CS = /usr/local/bin/tcpdump : PHYSICS = (operator) DUMP
```

```
lynda        ALL = (ALL) ALL, !SHELLS
```

```
%wheel       ALL, !Physics = NOPASSWD: PRINTING
```

sudoers discussion

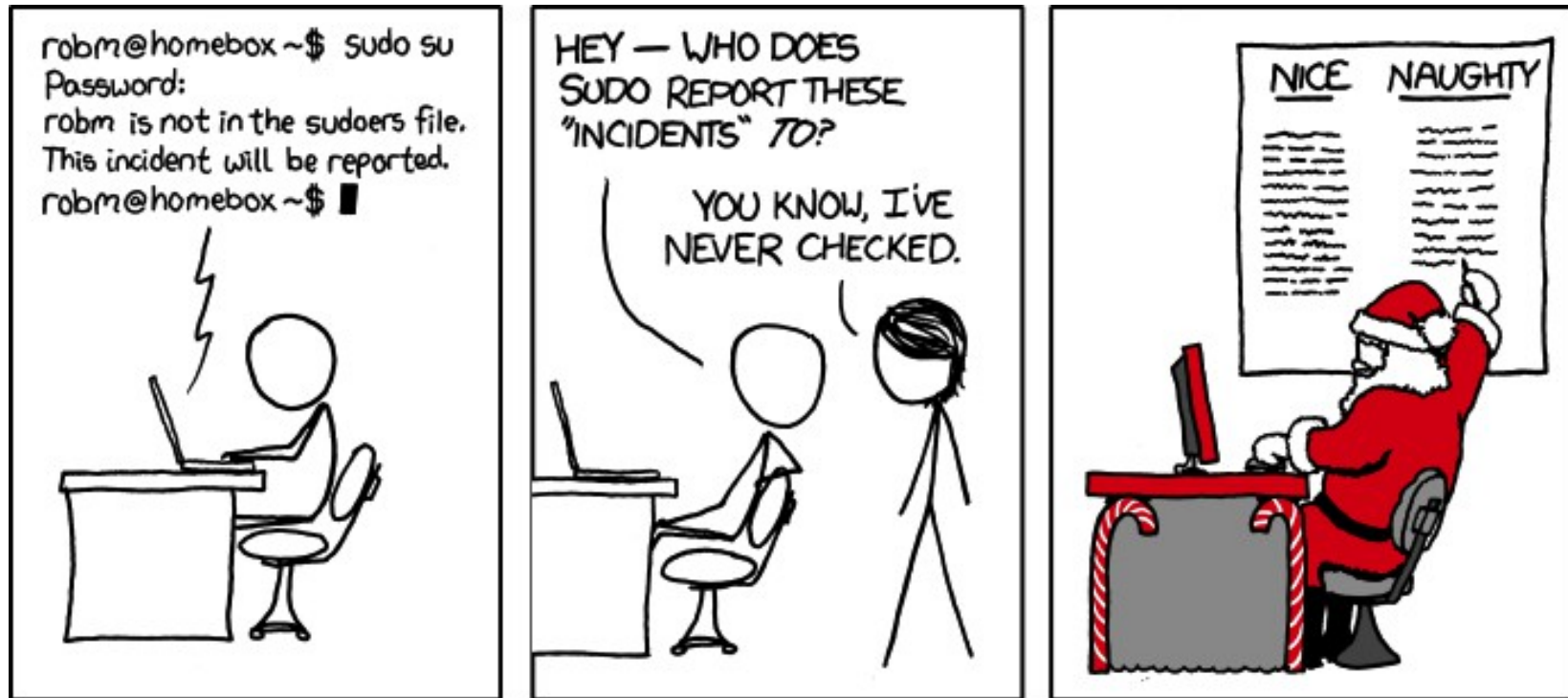
- Each permissions line includes
 - Users to whom the line applies
 - Hosts on which the line applies
 - Commands that the users can run
 - Users as whom the commands can be executed
- Use **visudo** to edit
 - If EDITOR environment variable set correctly
 - Locks file
 - Checks changes you made
- Example:

```
% sudo -u operator /sbin/dump 0u /dev/hda2
```

sudo advantages

- Accountability – commands are logged
- Operators can do chores without root privileges
- Real root password can be known to very few people
- **sudo** is faster to use than **su** or logging in as root
- Privileges can be revoked without changing root pw
- A complete list of users with root is maintained
- Less chance of a root shell being left unattended
- A single file can control access for an entire network

sudo logging



sudo disadvantages

- `/etc/sudoers` file is everything!
- Users with **sudo** privileges must protect their accounts as if they were root!
- Command logging can be avoided by starting a shell, or running some program that allows shell escapes



Other pseudo-users

- bin
 - Legacy owner of system commands
- daemon
 - Owner of unprivileged files and processes
- nobody
 - Account for remote roots of NFS systems
 - They often can't stay UID 0!
 - They need to be mapped to something

Group passwords

- The **newgrp** command allows a user to change the default group
 - Starts a new shell
 - If the group has a password, it will prompt for the password
 - Sometimes might give access, even if user not in list (varies)
- Group passwords are antiquated and not recommended
 - Must copy and paste password info
 - Group passwords are world readable
- RH/Fedora Linux has **gpasswd** command to set group password, put into `/etc/gshadow`, and more

sudo bang bang

