Essentials of Linux Systems Administration for Bioinformatics

José Héctor Gálvez

MonBUG

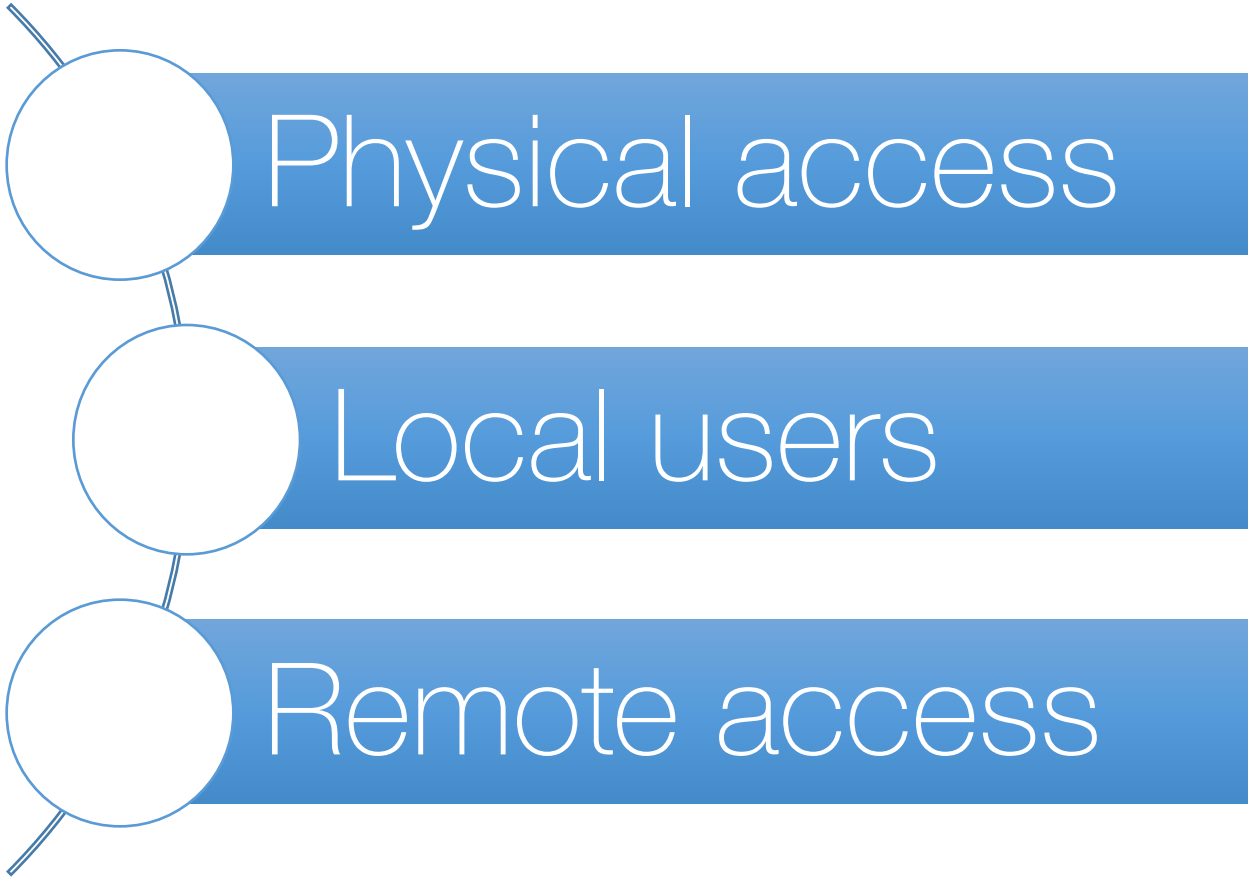
March 29, 2017

Identifying your role and responsibilities are the first steps when interacting with a new system

	Regular User	System Administrator
Security	✓	✓
Backups	✓	✓
Troubleshooting	✓	✓
Software installation	*	✓
User management	✗	✓
System updates	✗	✓

You might be the *de facto* system administrator of your lab's workstation!

Security should be the No. 1 concern of both users and system administrators



No system is 100% secure

- **The human factor is the weakest link**
- **Having a security policy will help prevent exploits**

- `passwd`
- `chmod`

Linux Security Modules:

- `selinux`

In bioinformatics and scientific computing regular backups are especially important

What to back up?

Raw data	✓✓✓
User files (`/home`)	✓✓
System configuration files	✓✓
Log files	✓
Software	*
`/tmp`	✗
Pseudo-filesystems/swap	✗

Treat data as *read-only*:

- **Only allow programs to read data and create new, separate files of results**
- **Always make back-ups**

- `rsync`
- `tar`, `dd`, `dump`, `restore`
- **RAID arrays (`fdisk`)**

All users should know the basics of troubleshooting and ask for assistance when necessary

1. **Characterize** the problem
2. **Reproduce** the problem
3. Always **try easy things** first
4. **Eliminate** possible causes, *one at a time*
5. Check system **logs**
   - `/var/log/messages`
   - `/var/log/secure`

Keep a checklist

For example:

- **Check exit status**
- **Check man pages**
- **Check file permissions…**

- **`kill, killall, pkill`**
- **`nice,unice`**
- **`ps,top,iostat`**

Bioinformatics requires the use of custom software, so having an installation policy is a good idea

- Package managers offer advantages
  - Automation
  - Scalability
  - Repeatability
  - Security
  - Auditing
- Other online repositories (github, bitbucket) can provide similar advantages

- `yum,APT`
- `make`
- `git clone`

Systems with several users require constant monitoring and limited permissions

- For regular users, consider:
  - Expiry dates
  - Memory quota/limited priority
  - Periodic password changes
  - Lock or disable inactive accounts
- Create user groups to manage several users at once
- Use the root account (or **`sudo`**) only when absolutely necessary

Never edit these files directly!

- `/etc/passwd`
- `/etc/group`
- `/etc/shadow`

- **`useradd,groupadd`**
- **`usermod,groupmod`**
- **`passwd,chage`**
- **`chmod,chown`**

An out-of-date system is less secure and might run slower than one that is regularly updated

- Regular schedule for updates
- Never skip security updates
- Package Management Systems:
  - Regular updates
  - “Smart-upgrade”
  - Verify package integrity
  - Remove packages
  - Clean cache

- `APT, yum`
- `RPM, DPKG`

A well-administered system is key for system reliability and research reproducibility

Further resources: