11

Answers to Even-numbered Exercises

2. How does the use of lock icons that require password authentication to open reflect the principle of least privilege?

   While restricting system administration tasks to certain users helps, many users are likely to use an administrative account all the time. Requiring authentication makes the system more secure in two ways. First, it ensures that users know when they are performing administrative tasks. Second, it prevents unauthorized users from walking up to a terminal logged in to an administrative account and performing administrative tasks.

4. How would you allow a user to execute privileged commands without giving the user the Superuser password?

   You can create a setuid program that belongs to a group that only the user who is to execute the program belongs to and that has no permissions for other users. Alternatively, you can implement sudo to grant the user permission to execute the file (see the sudo and sudoers man pages).

6. What is the difference between **LaunchAgents** and **LaunchDaemons**? Why does each user's home directory have a **Library/LaunchAgents** directory but not a **Library/LaunchDaemons** directory?

   LaunchAgents are run on behalf of a specific user; **LaunchDaemons** are systemwide. Users may have their own agents, which run only when they are logged in, but it makes no sense to talk about a systemwide service that runs only when a given user is logged in.

8. On Mac OS X Server, the **root** account starts out with the same password as the first user account created. Why is this less of a security flaw than it might seem to be?

   Because the first user account created can use sudo to obtain **root** privileges, anyone who has cracked the password on that account has **root** privileges anyway.

10. If you use launchctl to load a job when you do not have **root** privileges, a new copy of **launchd** is started. Why does Mac OS X not simply load your job into the copy of **launchd** that is already running as process 1?

    That copy of **launchd** is running as **root**, and only **root** has the privileges necessary to send it messages, such as causing **launchd** to load a new job. Running a new copy with your privileges is more secure than exposing the system **launchd** process to a non-**root** user.