Assignment 3

50 pts

This is a written assignment made up of eight problems. Your solution is due on October 17, 10:00 P.M. PDT. You may work with *one* other person in the class on this assignment. See Section 4 for additional information on submitting your solution. You and your partner may *not* discuss your solution with other students until seven days after the assignment deadline. You may consult any online references you wish. If you use any text in your answer that you did not write yourself, you *must* document that fact. Failure to do so will be considered a violation of the academic integrity policy.

1 Root Privileges

In the traditional Unix security model, processes running as the superuser have very few restrictions on what they are allowed to do. You decide to investigate.

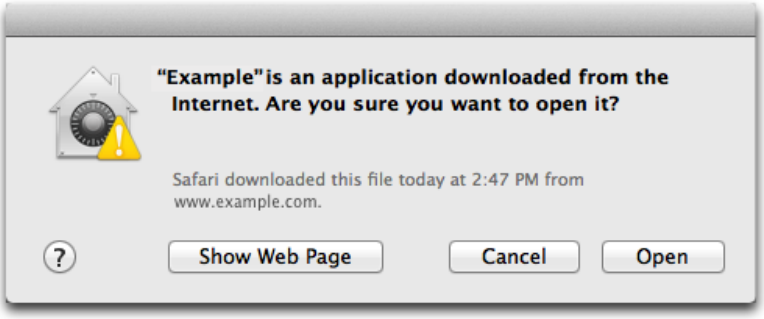
Problem 1: Running Processes. Identify two processes currently running as root on your Unix system. Why do they need to be running as root? The two processes must be different executables and may not be the `login` process mentioned in class.

Problem 2: Setuid root. Identify two executables on your system that are owned by root and have the setuid bit set, other than `passwd` and `sudo`, which were mentioned in class. Why do they need root privileges?

"Your Unix system" can be any Unix system you've used, including the ACMS computer lab machines or the VM image you used for Assignment 2.

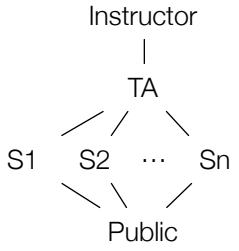
2 File Quarantine

Problem 3: Integrity or Secrecy. When you try to open an application you download from the Internet, Mac OS X shows the alert shown below. Is MacOS X concerned about the application's *integrity* or *confidentiality*?



3 CSE 127 Multilevel Security

In class we discussed a security lattice for CSE 127, shown at right, to be used with the Bell-LaPadula security model. Now consider the auto-grader process used to grade Assignment 2. The auto-grader reads the file you submitted, which are marked Si, where i is a number so that each student has his/her own label, and produces a report with your grade.



Problem 4: Pure BLP label. If the auto-grader is cleared up to Instructor level and reads no other information, what should be the security label assigned to the report produced for your solution according to the BLP model *without* the high-water mark principle?

Problem 5: BLP+HWM label. If the auto-grader is cleared up to Instructor level and reads no other information, what should be the security label assigned to the report produced for your solution according to the BLP model *with* the high-water mark principle?

Problem 6: Pure BLP, next student. After grading your solution, would the auto-grader process be allowed to open the solution of another student according to the BLP model *without* the high-water mark principle? If so, what would be the label assigned to the report? If not, why?

Problem 7: BLP+HWM, next student. After grading your solution, would the auto-grader process be allowed to open the solution of another student according to the BLP model *with* the high-water mark principle? If so, what would be the label assigned to the report? If not, why?

Now consider an auto-grader for a hypothetical future programming assignment where students are allowed (but not required) to work in pairs.

Problem 8: Group Lattice. Assuming that each group (1 or 2 students) submits a single file for grading.[1] Show a BLP security lattice that could be used for this scenario. You may attach a figure in JPEG or PNG format illustrating the lattice.

4 Submitting Your Solution

4.1 Solution Format

Your solution to this assignment must be a plain text file named "`{PID}-hw3.txt`" (where `{PID}` is your PID) containing your answers. The contents of the file must follow a specific format:

1. The first line of the file must be your name, last name first, with a comma between last name and first name.
2. The second line of the file must be your student id number.
3. The third line must be "`Assignment 3`".
4. If you worked with another student, the fourth line must be "`Worked with`" followed by a space and the name of your partner, last name first, with a comma between last name and first name. If you worked alone, the fourth line must be "`Worked alone`".
5. The fifth line must be blank.
6. Your solution to each problem must start with "`Problem` X" (where X is the problem number), with a blank line before and after "`Problem` X".

Figure 1 below shows an example of this format.

[1] As with this assignment, future assignments may require *each* student to submit their own solution even if he/she worked in a group. For the scenario in Problem 10, however, assume each group submits a single solution file.

```
Foster, Ian
A00000000
Assignment 3
Worked with Maskiewicz, Jacob

Problem 1

Are we part of the problem?

Problem 2

Or part of the solution?

...
```

Figure 1: Example solution file format.

4.2 Additional Files

You may include figures with your text answers. Figures must be named "{PID}-hw3fig{N}.{EXT}" where {PID} is your PID, {N} is the figure number, and {EXT} is either jpg or png, depending on file type. Combine your text file "{PID}-hw3.txt" and all figures into a single ZIP or tar+gzip archive. Then follow instructions below for encrypting and signing this file.

4.3 Encryption and Signature

Your solution must by submitted via email to cs127f1@ieng6.ucsd.edu by October 17, 2017, 10:00 P.M. PDT. Both members in a group must submit a solution; however, your answers to each problem may be the same as your partner's. The text file (if submit text only) or archive file (if including figures) must be signed with your PGP key and encrypted to the cs127f1@ieng6.ucsd.edu PGP key, which is provided on the CSE 127 Web page and has key fingerprint:

E1BF 1E04 1104 28DA 4F89 6543 B033 B3DC 10D3 7DBD.

You must send a plain email with an encrypted and signed file as an attachment. The email must have the subject "Homework 3 Submission" and the attachment must be named "{PID}-hw3.{EXT}.asc" (where {PID} is your PID and {EXT} is one of zip, tgz, or txt).

To sign and encrypt your submission with GPG, you can use the following command:

```
gpg --encrypt --sign --armor -r cs127f1@ieng6.ucsd.edu {PID}-hw3.{EXT}
```

This will produce a file named "{PID}-hw3.{EXT}.asc" in the same directory. You will need to have imported the cs127f1@ieng6.ucsd.edu public key into your GPG keyring first.