

# Information Security

## Homework 1

Dr. Edwin Sha

Due Date: **1pm, Feb 28, 2012**

The homework is important to help your learning. In this course, many homework questions do not have standard solutions. And many questions are intended to be vague. It is okay to have your own assumptions but make them reasonable. Remember to put down your assumptions in the report.

**PUT DOWN YOUR EMAIL ADDRESS IN YOUR REPORT.** Give me the hard-copy of your report. By the way, I don't like handwriting report. Please use some computer formatting tool such as LaTeX or Word. To save your time, you can draw figures by hand.

If you have questions, ask TA first. TA will grade your homework. It is YOUR responsibility to make your answers understandable. Some deduction of points will be given if your answers are not clear to TA.

**You must do it yourself.** You must not copy any solutions from others.

- (10 points) Consider a simple OS running in a standard **dual-mode** CPU. This simple OS allows user programs to jump directly to and execute any OS kernel service routines such as the one controlling hard-disk device controller without going through "trap" mechanism. Answer the following questions.  
(A) Can a user program directly control hard-disk device controller?  
(B) Why or why not?
- (10 points) Which of the following instructions should be *privileged* (a) Set value of timer. (b) Read the clock. (c) Set value of Stack Pointer in your program. (d) Turn off interrupts. (e) Trap instruction.
- (10 points) (A) How can the scheduler in OS be run if the current running process is in an infinite loop like *while(1);*?  
(B) Give two possible situations that this scheduler can be called.
- (10 points) Describe, in Unix, how a user process calls an I/O library function, say read data from a file in the hard-disk, through system calls. Explain the mode change, software interrupt, context switching briefly.
- (10 points) (A) Can we allow ordinary users to set mode bit to be kernel mode by running something like "set kernel-mode" instruction? Is it safe? Explain your reasons.  
(B) We know that kernel data are critical. We also know that the Trap instruction can make the mode be the kernel mode. Why is it hard for a hacker to use trap instructions to put itself to be in kernel mode and destroy kernel data?
- (10 points) An assembly statement "load Addr, Reg1" (load the data from Addr in the memory to Register 1.) is fetched and to be executed in CPU. Explain the possible situations during the translations of the virtual address and the data loading including a possible page fault. Assume this system uses TLB, one-level paging system, data cache.

7. (10 points) (A) A computer with a 32-bit address uses a one-level page table. Virtual addresses are split into a 20-bit page table field, and an offset. How large is a page and how many are there in the virtual address space? (B) A page table might be large. Explain clearly why a multiple page table scheme can reduce the required size of a page table. Use a 2-level page system to explain your points. You may draw any figure by hand. (C) To load or store a page table is time consuming. Does each process has its own page table or a whole system shares one page table? Why or why not?
8. (10 points) Alice knows that the beginning part of memory is critically used for the **Linux** kernel in a Pentium PC, so she wrote a small assembly program to declare a buffer from memory address 0 to 999,999 and reset these 1 million bytes to be 0. Please know that this program does not do anything extra related to Linux kernel.
- (A) If Alice is an ordinary user, after she runs the program, will the system crash? Why?  
 (B) If Alice is the “root” (administrator), after she runs the program, will the system crash? Why?
9. (10 points) We know that each file has a *set-user-id* bit. If it is on, a so-called *SETUID* function will be performed. Describe how this SETUID works so that an ordinary user can modify the */etc/passwd* file, which is owned by *root*, by using *passwd* program, but not by using other programs, say *vi*, or *emacs*.
10. (10 points) In class I described a security flaw in TENEX operating system. TENEX used passwords to protect files. The operating system checked password one character at a time, stopping as soon as it saw that the password is wrong. By using the page fault trap, it is possible to find a correct 8-character password with  $8 \times 128$  checking steps assuming each character is an ASCII code. This gives a much less running time to find a password than the exhaustive search with  $128^8$  steps. How to modify the password checking algorithm so that this bug can be fixed. Give me two possible solutions.
11. (10 points) Assume that your current directory in UNIX is *A*. You perform the following UNIX command: *ln foo ../B/newfoo*. Explain how UNIX performs this “ln ” command. You should explain clearly based on the I-node and the directory files *A*, *B*. Your explanation should include the security check. Assume that you belong to the “others” for each permission check.
- Make sure you at least answer the following: Is there a new I-node being created? Any data related to this new I-node? What are changed in directory files *A* and *B*? What need to be checked for directory files *A*, *B* and file *foo* assuming that you belong to “others”? At least consider the cases that the permission bits for *A* and *B* are either 755 or 733.
12. (15 points) Read the papers about Internet worm posted in the course web page. It might be hard for anyone to understand the details of the worm, but it certainly gives you a good idea how the worm works. Recently notorious worms still used the same idea developed in this worm so this is definitely the most important worm to study! Moreover, the password cracking method is simple and effective. To make sure that you read the papers, you should answer the following questions.

- (a) Briefly describe how the worm penetrates into another computer system and propagates itself.
  - (b) What are the input parameters of the bootstrap program? What are their purposes?
  - (c) Why the main program of the worm makes the zeroth argument to be “sh”? Why it forks a child process repeatedly?
13. (10 points) Probability plays an extremely important role in security. So you should be familiar with the basic theory of probability. Assume that you have a fair coin. After you toss the coin, the probability of “head” is 0.5 and the probability of “tail” is also 0.5, so this is called a fair coin. Assume that each toss is independent. What is the probability that the head occurs exactly once after you toss  $n$  times? What is the probability that the  $n + 1$ th toss is head given that the previous  $n$  tosses only had one head?

**No cheating is allowed.** I will be very upset if we find any cheating. You must do the homework by yourself.