Chapter 3 - Programs & Programming

**1. Suppose you are a customs inspector. You are responsible for checking**

**suitcases for secret compartments in which bulky items such as jewelry might be**

**hidden. Describe the procedure you would follow to check for these**

**compartments.**

**2. Your boss hands you a microprocessor and its technical reference manual.**

**You are asked to check for undocumented features of the processor. Because of**

**the number of possibilities, you cannot test every operation code with every**

**combination of operands. Outline the strategy you would use to identify and**

**characterize unpublicized operations.**

**3. Your boss hands you a computer program and its technical reference manual.**

**You are asked to check for undocumented features of the program. How is this**

**activity similar to the task of the previous exercises? How does it differ? Which**

**is the more feasible? Why?**

**4. A program is written to compute the sum of the integers from 1 to 10. The**

**programmer, well trained in reusability and maintainability, writes the program**

**so that it computes the sum of the numbers from k to n. However, a team of**

**security specialists scrutinizes the code. The team certifies that this program**

**properly sets k to 1 and n to 10; therefore, the program is certified as being**

**properly restricted in that it always operates on precisely the range 1 to 10. List**

**different ways that this program can be sabotaged so that during execution it**

**computes a different sum, such as 3 to 20.**

**5. One way to limit the effect of an untrusted program is confinement:**

**controlling what processes have access to the untrusted program and what**

**access the program has to other processes and data. Explain how confinement**

**would apply to the earlier example of the program that computes the sum of the**

**integers 1 to 10.**

**6. List three controls that could be applied to detect or prevent off-by-one errors.**

**7. The distinction between a covert storage channel and a covert timing channel**

**is not clearcut. Every timing channel can be transformed into an equivalent**

**storage channel. Explain how this transformation could be done.**

**8. List the limitations on the amount of information leaked per second through a**

**covert channel in a multiaccess computing system.**

**9. An electronic mail system could be used to leak information. First, explain**

**how the leakage could occur. Then, identify controls that could be applied to**

**detect or prevent the leakage.**

**10. Modularity can have a negative as well as a positive effect. A program that is**

**overmodularized performs its operations in very small modules, so a reader has**

**trouble acquiring an overall perspective on what the system is trying to do. That is,**

**although it may be easy to determine what individual modules do and what small**

**groups of modules do, it is not easy to understand what they do in their entirety as a**

**system. Suggest an approach that can be used during program development to**

**provide this perspective.**

**11. You are given a program that purportedly manages a list of items through hash**

**coding. The program is supposed to return the location of an item if the item is**

**present or to return the location where the item should be inserted if the item is not in**

**the list. Accompanying the program is a manual describing parameters such as the**

**expected format of items in the table, the table size, and the specific calling sequence.**

**You have only the object code of this program, not the source code. List the cases**

**you would apply to test the correctness of the program’s function.**

**12. You are writing a procedure to add a node to a doubly linked list. The system on**

**which this procedure is to be run is subject to periodic hardware failures. The list**

**your program is to maintain is of great importance. Your program must ensure the**

**integrity of the list, even if the machine fails in the middle of executing your**

**procedure. Supply the individual statements you would use in your procedure to**

**update the list. (Your list should be fewer than a dozen statements long.) Explain the**

**effect of a machine failure after each instruction. Describe how you would revise this**

**procedure so that it would restore the integrity of the basic list after a machine**

**failure.**

**13. Explain how information in an access log could be used to identify the true**

**identity of an impostor who has acquired unauthorized access to a computing system.**

**Describe several different pieces of information in the log that could be combined to**

**identify the impostor.**

**14. Several proposals have been made for a processor that could decrypt encrypted**

**data and machine instructions and then execute the instructions on the data. The**

**processor would then encrypt the results. How would such a processor be useful?**

**What are the design requirements for such a processor?**

**15. Explain in what circumstances penetrate-and-patch is a useful program**

**maintenance strategy.**

**16. Describe a programming situation in which least privilege is a good strategy to**

**improve security.**

**17. Explain why genetic diversity is a good principle for secure development. Cite an**

**example of lack of diversity that has had a negative impact on security.**

**18. Describe how security testing differs from ordinary functionality testing. What**

**are the criteria for passing a security test that differ from functional criteria?**

**19.**

**(a) You receive an email message that purports to come from your bank. It**

**asks you to click a link for some reasonable-sounding administrative**

**purpose. How can you verify that the message actually did come from your**

**bank?**

**(b) Now play the role of an attacker. How could you intercept the message**

**described in part (a) and convert it to your purposes while still making both**

**the bank and the customer think the message is authentic and trustworthy?**

**20. Open design would seem to favor the attacker, because it certainly opens the**

**implementation and perhaps also the design for the attacker to study. Justify that open**

**design overrides this seeming advantage and actually leads to solid security.**