CSC 405 Introduction to Computer Security

Homework Assignment #2 Solutions

2. (20 points) 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.

Note: This is an open question; there is no strict answer for this question.

Use the technical manual to make a list of the microprocessor's operation codes and what their expected range of operands should be. For each operation code, randomly choose a few operands to test, some within the expected range of operands, some at the extreme limits of the range and finally pick a few that are completely outside the range.

3. (15 points) 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 exercise? How dose it differ? Which is the more feasible? Why?

Note: This is an open question; there is no strict answer for this question.

We can use the same method in previous problem to test the computer program.
For computer program, we know the source code or binary code. So we can use other methods, such as static analysis, to test it. In other words, we can use white box testing or gray box testing. But for the hardware, we can only use black box testing.
Testing a computer program is more feasible, because we can use more methods.

5. (20 points) A program is written to computer 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 scrutinized the code. The team certified that the program properly set k to 1 and n to 10; therefore, the program is certified as being properly restricted in that it always operate 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, for example, 3 to 20.

a) Someone changes the source code before its complication,
b) Someone patches (i.e.,) the binary object code while it is stored on disk before execution,
c) During execution, an outside process patches the object code.

6. (15 points) One means of limiting 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.

Assuming the only activity of the program is computing the sum from 1 to 10, confinement would achieve two things. First, the confining program would act as a filter between the callers and the untrusted program. A calling program would call the confining process, requesting to call the summation program. The calling program would have no direct access to the summation program. Second, the confining program would check the result to the summation program. In this simple situation, the confining process could check that the answer was exactly 55 (the sum from 1 to10). In a more realistic situation, the confining process could check the computation for reasonableness: considering the magnitude of the input values, values of other system variables, the name or owner of the calling program, etc., is the result reasonable? Are the requests for access to auxiliary system resources by the untrusted program reasonable?

Confining programs such as described here do exist. They are generally called "wrappers" because they wrap the untrusted code in a trustworthy filter

7. (15 points) List three controls that could be applied to detect or prevent salami attacks.

Examples of controls: (a) program development controls, in which the code of a program is rigorously scrutinized for improper activity, (b) spot checks of random accounts, involving recomputation of answers by hand to verify that correct amounts are being credited, and (c) requirement of matching totals for many intersecting subsets of accounts

8. (15 points) The distinction between a covert *storage* channel and a covert *timing* channel is not clear-cut. Every timing channel can be transformed into an equivalent storage channel. Explain how this transformation could be done.

Covert channels typically require access to a shared clock to time when bits become available in the covert resource and when bits can be replaced. Thus, even with pure storage channels, there is an element of timing.

A covert timing channel works by modulating the time at which something occurs. But the something (which might be an interrupt or access to the CPU or unlocking a semaphore, for example) is itself a resource (the interrupt, the processing, or the semaphore), represented by a storage table entry. Thus, the table entry or the something itself becomes the shared resource visible to the two cooperating processes form which the covert channel is built.