Information Survivability, Security, and Fault Tolerance

Matt Bishop
Department of Computer Science
University of California at Davis
Davis, CA 95616-8562

A *survivable* computer system is defined as "one that continues to operate even after one or more of its components fail" ([1], p. 453). Adapting this definition to information gives that *information survivability* is the ability to retain access to information even after one or more of the access paths or mechanisms fail. The problem is broader than data stored on computer; for example, if the Census Bureau computers break down, the information survives because it is saved in books and on printouts. For our purposes, we assume the information is stored on computer, and accessed only through the system or over a network.

Unlike traditional fault tolerant computing, fault tolerant computing in support of information survivability must provide access to information in the face of attacks on integrity and availability. One large class of fault tolerant techniques uses a quorum of redundant components to determine what the "correct" information is; examples are triple modular redundancy and *N*-version programming. Such a technique implicitly assumes that some distinguished elements of the power set of the redundant components are trustworthy. In a quorum system, the distinguished sets are those that contain a majority of components. These techniques assume that a majority of components will function correctly. In an attack involving the induction of deliberate faults in these components, this assumption fails.

The key to understanding the security of any computer system is to understand the assumptions made (explicitly and implicitly) about the system. The techniques of computer security can provide information leading the designer or analyst to determine which sets of components are least likely to fail, and to augment components to minimize the probability of their failing.

The problem of information survivability encompasses computer security, which is itself a problem of fault tolerance and avoidance. An attack is a deliberate fault ("the mechanical or algorithmic cause of an error", [2], p. 126) that disrupts service, causing errors. Designing and implementing to prevent attacks is simply fault avoidance. Designing and implementing to allow correct functioning in the face of attacks is fault tolerance. More research in this specific area might offer new techniques for handling attacks. Specifically, how can one keep a system secure when part of it are corrupted during an attack? This fruitful area has been explored only in the context of denial of service. It should be explored in the areas of integrity (including authentication) and confidentiality.

[1] Dietel, Harvey M., *An Introduction to Operating Systems* (Revised First Edition), Addison-Wesley, Reading, MA (1984).

[2] Randell, B., Lee, P. A., and Treleaven, P. C., "Reliability Issues in Computing System Design," *ACM Computing Surveys* **10**(2) pp. 123–166 (June 1978).