Panel: Issues in Software Engineering for Survivable Systems

Nancy R. Mead, Panel Chair
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213 USA
+1 412 268 5756
nrm@sei.cmu.edu

ABSTRACT
Network systems are a primary enabling agent in business, industry, government, and defense. This panel session addresses software engineering issues for survivable systems, where survivability is defined as the capability of a network system to complete its mission in a timely manner, even if significant portions are incapacitated by attack or accident.

Keywords
Survivable systems, software engineering

1 NETWORK SYSTEMS SURVIVABILITY
Network systems are a principal enabling agent in business, industry, government, and defense. Major economic sectors all depend on a vast array of networks operating on local, national, and global scales. This pervasive societal dependency on networks magnifies the consequences of failure and amplifies the critical importance of ensuring their survivability.

Survivability is the capability of a system to complete its mission in a timely manner, even if significant portions are incapacitated by attack or accident. In particular, survivability refers to the capability of a system to provide essential services in the presence of successful intrusion and to recover compromised services after intrusion occurs. Effective definition of software engineering techniques for such systems is an essential first step in their development.

Traditional security measures have focused on defense against intrusion, recognition of intrusion, and application of recovery strategies after the fact. Today's distributed systems can no longer depend on isolation as the sole defense against intrusion. Survivable systems need a more proactive approach, so that in addition to resistance and recognition techniques, systems are able to recover and continue to perform essential tasks. There has been little attention to incorporating these techniques into good software engineering practice. Security is often treated as an add-on patch to existing systems that were developed without regard to security and survivability.

This panel will focus on the distinction between the software engineering techniques that we see in today's secure systems and the software engineering techniques that will be needed for unbounded, survivable systems. New techniques will be needed to allow the system to recognize, respond, and recover from intrusion. Software engineering aspects of survivability will need to be addressed in a proactive rather than reactive manner.

2 PANELIST POSITION STATEMENTS
John C. Knight, University of Virginia
The breadth of society's dependence on information systems is quite surprising. In the banking system, for example, many different financial services, such as support for equities markets and the payment and foreign exchange systems, are dependent on information systems that operate together. Similarly, in power generation and control, different service areas are tightly integrated as power flows between them; this forces integration of their control systems.

New and inexpensive computer hardware offers the opportunity to enhance existing applications in innovative ways and develop new applications. In the freight-rail industry, for example, tremendous improvements in efficiency are being achieved by the implementation of "just-in-time" delivery that is made possible by fast computers and extensive communications capabilities. Many factories no longer maintain warehouses of parts for manufacturing because they depend on parts being delivered as they are needed.

The loss of the services that these systems provide could be very serious. Some of the consequences of failure are fairly obvious, such as in the air traffic control system. Other consequences of failure are less obvious. The failure of freight-rail service would disrupt many aspects of manufacturing, energy supply, and food distribution.

A factor that complicates the situation is the interdependence of some of these applications. Thus, for example, management of transportation systems will be affected sig-
nificantly if there is a widespread loss of power. Similarly, loss of communication service will disrupt many other information systems such as finance, electronic commerce, and transportation.

The issues in this area are significant and urgent. The core technology at the heart of critical information systems is software, and the software engineering research community faces many challenges if adequate availability and security of these systems are to be achieved.

Richard C. Linger, Software Engineering Institute

The enabling environment for survivability of critical infrastructure systems will likely include regulatory and standards components, as well as technical and engineering components. Much survivability analysis of current systems is based on business risks; societal risks involve broader consequences and will become increasingly prominent. While survivability can never be guaranteed, improved development technologies and methods can explicitly address survivability to help reduce, quantify, and manage the risks involved. As an engineering objective, survivability impacts all aspects of software engineering.

In particular, new techniques are required for specifying survivability requirements; developing survivable architectures; improving intrusion resistance, recognition, and recovery strategies; analyzing and measuring survivability; increasing survivability of legacy systems; and testing and assuring system survivability. These techniques in turn require improvements in understanding the underlying behavioral semantics of large-scale, highly distributed network systems.

The Software Engineering Institute (SEI) is working on a Survivable Network Analysis method and a Survivable Systems Simulator (to investigate emergent and self-stabilizing behavior). The SEI has conducted two Information Survivability Workshops, to bring infrastructure domain experts and survivability researchers together, and has initiated research with the Carnegie Mellon University (CMU) School of Computer Science on semantic foundations for survivability of network systems. The SEI is also working with the CMU Heinz School of Public Policy and Management to offer a masters-level security/survivability specialization.

Robyn R. Lutz, Jet Propulsion Laboratory/CIT

Experience with robust, safety-critical systems offers some approaches to meeting four key challenges in the development of survivable systems:

Evolving requirements: One of the most difficult aspects of engineering survivable systems is the degree to which the requirements for survivability evolve during the system's development. Survivability requirements change with advances both in defensive measures and in threats. Component-based development, with reuse of product families where appropriate, supports rapidly evolving systems.

Difficult design tradeoffs: As an example, certifiable commercial off-the-shelf (COTS) components with formally specified interfaces enhance the predictability of composed behavior. However, the lack of diversity in reusable components may increase the system's openness to attack. Explicit requirements negotiations among stakeholders, and explicit documentation of operating assumptions and limits of survivability, assist in tradeoff decisions.

Adequate hazard analysis: For survivable systems, hazard analysis can reduce performance risk and help structure the on-going process of refining and prioritizing the survivability requirements. Some hazards cannot be avoided or prevented, but must be handled in the software, increasing its complexity. Software failure modes and effects analysis, and software fault-tree analysis enhance understanding of interactions and the contributing causes of hazards.

Verification of new architectures: These architectures are inadequately tested by traditional verification techniques. Formal methods offer a way to begin modeling and investigating the behavior of the planned system, and to validate that key properties hold invariantly in the system as modeled.

Jeffrey Voas, Reliable Software Technologies

It is nearly impossible to punish the perpetrators of crimes against digital systems for the following reasons: identifying the individuals is extremely difficult and such crimes can originate thousands of miles away. Therefore traditional deterrents such as retaliation or other punishments are infeasible. The only sure form of protection is absolute defense.

Absolute defense begins with policies that define which services should not be allowed to operate over publicly accessible networks. It is foolhardy to believe that we will solve the numerous "security" problems created by integrated computer networks in the near future. We are always a step behind. We must accept this, quantify risks, and from there devise security schemes for the "worst consequence" scenarios.

This does not mean that we should not seek long-term technical improvements in the sciences of intrusion detection, firewalls, etc. But it does mean that in the short-term, we should concentrate on public policies that protect those services (e.g., electricity) that cannot be compromised. If that means transferring such services from public to proprietary networks then so be it. This counters the current direction toward COTS offerings. But then our only other option is to hope for the best.

Unfortunately, politicians and legislation may be the best strategy for averting short and near-term disasters to the critical national infrastructure. Foolproof technical solutions to outsider attacks are not on the near-term horizon. Catastrophic service disruptions are.