The Survivability Imperative: Protecting Critical Systems

By Richard C. Linger, Robert J. Ellison, Thomas A. Longstaff, and Nancy R. Mead
Software Engineering Institute

The success of virtually all organizations in defense, government, and business is dependent on availability and correct functionality of large-scale networked information systems of remarkable complexity. Because of the severe consequences of failure, organizations are focusing on system survivability as a key risk management strategy. The Survivable Network Analysis (SNA) method provides a systematic means to assess and improve system survivability for risk reduction. Survivability can also be integrated into requirements definition for new or evolving systems.

Progress Demands System Survivability

Modern society is increasingly dependent upon large-scale, highly distributed systems that operate in unbounded network environments. Such systems improve efficiency by permitting entire new levels of organizational integration, but they also introduce elevated risks of intrusion and compromise. These risks can be mitigated within the organization's system by incorporating survivability capabilities.

Unbounded networks such as the Internet have no central administrative control and no unified security policy. Furthermore, the number and nature of nodes connected to such networks cannot be fully known. Despite the best efforts of security practitioners, no amount of hardening can assure that a system connected to an unbounded network will be invulnerable to attack.

The discipline of survivability can help ensure that systems can deliver essential services and maintain essential properties including integrity, confidentiality, and performance despite the presence of intrusions. Unlike traditional security measures, which often depend on central control and administration, survivability is intended to address network environments where such capabilities may not exist.

Survivability is defined as the capability of a system to fulfill its mission in a timely manner, even in the presence of attacks, failures, or accidents. As an emerging discipline, survivability builds on related fields of study, including security, fault tolerance, safety, reliability, reuse, performance, verification, and testing; moreover, it introduces new concepts and principles [1, 2, 3, 4, 5]. Survivability focuses on preserving essential services in unbounded environments, even when systems are penetrated and compromised.

In defining survivability, the term mission refers to high-level organizational objectives. Missions are not limited to military settings; any successful organization or project must have a vision of its objectives, whether expressed implicitly or as a formal mission statement. Judging mission fulfillment is typically made in the context of external conditions that affect achievement of mission objectives.

For example, a financial system may shut down for 12 hours during a period of widespread power outages caused by a hurricane. If the system preserves integrity and confidentiality of data and resumes essential services following the period of downtime, it can reasonably be judged to have fulfilled its mission. However, if the system shuts down unexpectedly for 12 hours under normal conditions or minor environmental stress and deprives users of essential financial services, it can be judged to have failed its mission, even if integrity and confidentiality are preserved.

Glossary of Survivability Terms

Accidents—A broad range of randomly occurring and potentially damaging events such as natural disasters. Accidents are often externally generated events.

Adaptation services—Survivable system functions provided to continually improve a system's capability to deliver essential services, typically by improving resistance, recognition, and recovery capabilities.

Attack—A series of steps taken by an intelligent adversary to achieve an unauthorized result. Attacks include intrusions, probes, and denials of service.

Essential services—Services that must be provided to system users even in the presence of attacks, failures, or accidents.

Failure—A potentially damaging event that results from deficiencies in a system or in an external element on which the system depends. Failures may be due to results from software design errors, hardware degradation, human errors, or corrupted data.

Recognition services—Survivable system functions that must detect attempted and successful attacks.

Recovery services—System functions to support the restoration of services after an attack. Recovery services also contribute to a system's ability to maintain essential services during an attack.

Resistance services—System functions that repel attacks and make them difficult and costly.

Survivability—A system's capability to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.

Unbounded network—A computer system or systems characterized by distributed administrative control without central authority, limited visibility beyond the boundaries of local administration, and lack of complete information about the network.

Timeliness is typically a critical factor in mission objectives, and is explicitly included in the definition of survivability. The terms attack, failure, and accident include all potentially damaging events; however, these terms do not partition events into mutually exclusive or even distinguishable sets. It is often difficult to determine if a particular detrimental event is the result of a malicious attack, a component failure, or an accident. Even if the cause is eventually determined, the critical immediate response cannot depend on speculations about the cause.

Attacks are potentially damaging events orchestrated by an intelligent adversary. Attacks include intrusions, probes, and denials of service. Moreover, the threat of an attack can have as severe an impact on a system as an actual occurrence. A system that assumes an overly defensive position because of an attack threat may significantly reduce functionality and divert excessive resources to monitoring the environment and protecting system assets.

Failures are potentially damaging events caused by deficiencies in a system or in an external element upon which the system depends. Failures may be due to software design errors, hardware degradation, human errors, or corrupted data.

Accidents describe a broad range of randomly occurring and potentially damaging events, such as natural disasters, that usually originate outside a system.

With respect to survivability, a distinction between an attack and failure or accident is less important than the impact of the event. It is often not possible to distinguish between intelligently orchestrated attacks and unintentional or random detrimental events. Survivability concentrates on the effect of a potentially damaging event. For a system to survive, it must recover from a damaging effect long before the underlying cause is identified. In fact, recovery must be successful whether or not the cause is ever determined.

It is important to recognize that mission fulfillment must survive—not any particular subsystem or component. The core concept of survivability is the capability of a system to fulfill its mission, even if significant portions of the system are damaged or destroyed.

Survivable Network Analysis

The SNA method depicted in Figure 1 was developed by the SEI Computer Emergency Response Team (CERT) Coordination Center as a practical engineering process for systematic assessment of survivability properties of proposed systems, existing systems, and modifications to existing systems [6, 7]. SNA is carried out at the architecture level as a cooperative project by an SEI team working with system architects, developers, and stakeholders. The method proceeds through a series of joint working sessions, culminating in a briefing on findings and recommendations. In this article, the focus is on attacks, although the trace-based, compositional SNA method applies to analysis of failures and accidents as well.

The SNA method provides a means for organizations to understand survivability in the context of their operating environments. What functions must survive? What intrusions could occur? How could intrusions affect survivability? What are the risks to the mission? How could architecture modifications reduce the risks? Systematic consideration of these questions through SNA reveals the risks and leads to mitigation strategies. Steps in the SNA method are defined as follows:

Step One: System Definition

The first step focuses on understanding mission objectives, requirements for the current or candidate system, structure and properties of the system architecture, and risks in the operational environment.

Step Two: Essential Capability Definition

Once step one is complete, essential services (services that must be maintained during attack) and essential assets (assets whose integrity, confidentiality, availability, and other properties must be maintained during attack) are identified, based on mission objectives and the consequences of failure. Essential service and asset uses are characterized by usage scenarios, which are traced through the architecture to identify essential components whose survivability must be ensured.

Step Three: Compromisable Capability Definition

Next, intrusion scenarios are selected based on assessment of environmental risks and intruder capabilities. These scenarios are likewise mapped onto the architecture as execution traces to identify corresponding compromisable components (components that could be penetrated and damaged by intrusion). In essence, intruders are treated as simply another class of users, and the design task for intrusion usage is to make it as difficult and costly as possible.

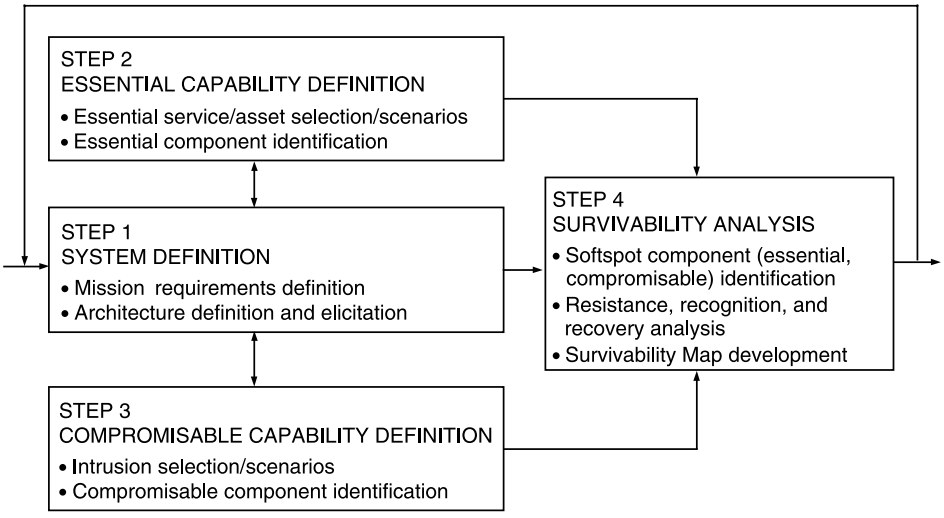
Step Four: Survivability Analysis

The final step of the SNA method takes aim at soft spot components of the architecture. These are components that prove both essential and compromisable, based on the results of steps two and three. Soft spot components and supporting architecture are then analyzed for the key survivability properties of resistance, recognition, and recovery (the three Rs), as well as for adaptation and evolution.

Resistance is the capability of a system to repel attacks. Recognition is the system's capability to detect attacks as they occur and to evaluate the extent of damage and compromise. Recovery, a hallmark of survivability, is the capability to maintain essential services and assets during attack, limit the extent of damage, and restore full services following attack. Table 1 depicts some strategies for improving survivability.

The analysis of the "three R's" is summarized in a Survivability Map as depicted in Figure 2. The map enumerates, for every intrusion scenario and its corresponding soft spot effects, the current and recommended architecture strategies for resistance, recognition, and recovery. The Survivability Map provides feedback about the original architecture and system requirements, and gives management a roadmap for survivability evaluation and improvement. In addition, survivability analysis often results in recommendations for security and survivability policy definition or modification. The SNA method has been applied to a number of systems with good results.

Figure 1. *The Survivable Network Analysis Method*



Key Property	Description	Examples
Resistance to attacks.	Strategies for repelling attacks.	System and user authentication, access control, encryption, fire walls, proxy servers, strong configuration management, dispersion of data, diversification of programs, application of system upgrades for known vulnerabilities.
Recognition of attacks and extent of damage.	Strategies for detecting attacks (including intrusions) and understanding the current state of the system, including evaluating the extent of the damage.	Recognition of intrusion usage patterns, virus scans, internal integrity checking, auditing, system configuration and network monitoring.
Recovery of full and essential services after attack.	Strategies for restoring compromised information or functionality limiting the extent of damage, maintaining or, if necessary, restoring essential services within the time constraints of the mission, restoring full service as conditions permit.	Restoration of data and programs, use of alternative services, operational procedures to restore system configurations, isolation of damage, ability to operate with reduced services or reduced user community.
Adaptation and *evolution* to reduce effectiveness of future attacks.	Strategies for improving system survivability based on knowledge gained from intrusions.	Incorporation of new patterns for intrusion recognition, adaptive filtering and logging.

Table 1. *Some Strategies for Improving System Survivability*

Customers have benefited from survivability improvements to system architectures, as well as from clarified requirements and early problem identification. Survivability is also the subject of ongoing research, as described, for example, in Fisher [8].

Adding Survivability to System Requirements

Survivability properties can also be integrated into the requirements definition for new or evolving systems [9]. Figure 3 depicts an iterative model for defining survivable system requirements. Survivability must address not only requirements for software functionality, but also requirements for software usage, development, operation, and evolution. Thus, five specific types of requirements definitions are relevant to survivable systems in the model of Figure 3, as discussed below.

System/Survivability Requirements

In this discussion, system requirements refers to traditional user functions that a system must provide. For example, a network management system must provide user functions for monitoring network operations, adjusting performance parameters, and so forth. System requirements also include non-functional aspects, such as timing, performance, and reliability. Survivability requirements refer to system capabilities for the delivery of essential services in the presence of attacks and intrusions, and recovery of full services.

Survivability requires that system requirements be organized into essential services and non-essential services, perhaps in terms of user categories or business criticality. Essential services must be maintained even during successful intrusions; non-essential services are to be recovered after intrusions have been dealt with.

Essential services may be further stratified into levels with each embodying fewer and more vital services as a function of increasing severity and duration of intrusion. It is also possible that the set of essential services may vary in a more dynamic manner depending on a particular attack scenario and the resulting situation. In this case, services that are essential under one scenario may not be essential under another resulting in different combinations of essential services that are scenario-dependent.

Thus, definitions of requirements for essential services must be augmented with appropriate survivability requirements. As shown in Figure 3, survivable systems may also include legacy and COTS components not originally developed with survivability as an explicit objective. Such components may provide both essential and non-essential services and may engender special functional requirements for isolation and control through wrappers and filters to help permit safe use in a survivable system environment.

Beyond functional requirements, survivability itself imposes new types of requirements on systems for resistance to, recognition of, and in particular, recovery from intrusions and compromises. A variety of existing and emerging survivability strategies, noted in Table 1 support these survivability requirements.

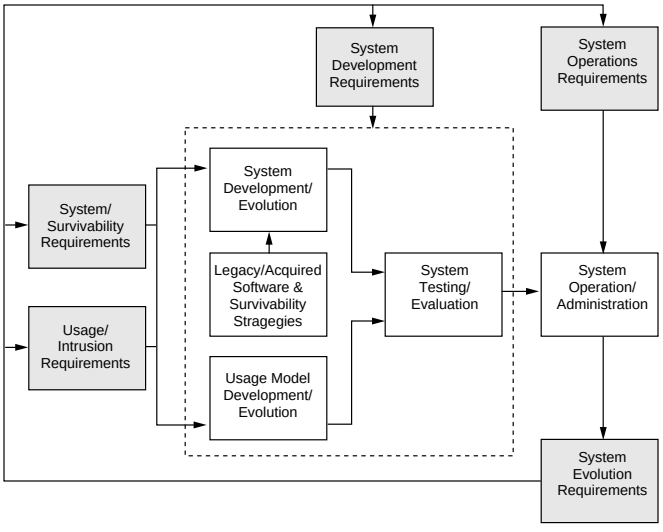
Survivable systems are envisioned as capable of adapting their behavior, function, and resource allocation in response to intrusions. When necessary, for example, functions and resources devoted to non-essential services could be reallocated to the delivery of essential services and intrusion resistance, recognition, and recovery. Requirements for such systems must specify the behavior for adaptation and reconfiguration in response to intrusions.

Systems can exhibit large variations in survivability require-

Figure 2. *Sample Survivability Map Format*

Intrusion Scenario	Softspot Effects	Architecture Strategies for:	Resistance	Recognition	Recovery
(Scenario 1) …		Current Recommended			
(Scenario n)		Current Recommended			

Figure 3. *Requirements Definition for Survivable Systems*



ments. Small local networks may have few or even no essential services with acceptable manual recovery times measured in hours. Large-scale networks of networks may be required to maintain a core set of essential services with automated intrusion detection and recovery times measured in minutes. Embedded command and control systems may require essential services to be maintained in real time, with recovery periods measured in milliseconds. Attainment and maintenance of survivability consumes resources in system development, operation, and evolution. Survivability requirements for a system should be based on costs and risks to an organization associated with loss of essential services.

Usage/Intrusion Requirements

Survivable system testing must demonstrate the performance of essential and nonessential system services, as well as the survivability of essential services during an intrusion. Because system performance in testing (and operation) depends totally on the usage to which it is subjected, an effective approach to survivable system testing is based on usage scenarios derived from usage models.

Usage models are developed from usage requirements, which specify legitimate usage environments and all possible usage scenarios. Usage requirements for essential and nonessential services must be defined in parallel with system and survivability requirements. Furthermore, intrusion usage must be treated on a par with legitimate usage and intrusion requirements, which specify that intrusion usage environments and all possible scenarios of intrusion use must be defined as well. In this approach intrusion usage is modeled in conjunction with the legitimate use of system services. Intruders may engage in usage scenarios beyond legitimate scenarios, but may also employ legitimate usage for purposes of intrusion if they become privileged to do so.

Development Requirements

Survivability places stringent requirements on system development and testing practices. Software errors can have a devastating effect on survivability and provide ready opportunities for intruder exploitation. Sound engineering practices are required to create survivable software. The following five principles—four technical and one organizational—are example requirements for survivable system development and testing practices:

- Precisely specify required functions in all possible circumstances of use.
- Verify correct implementations with respect to function specifications.
- Specify function usage in all possible circumstances of use, including intruder usage.
- Test and certify based on function usage and statistical methods.
- Establish permanent readiness teams for system monitoring, adaptation, and evolution.

Sound engineering practices are required to deal with legacy and COTS software components as well.

Operations Requirements

Survivability also places demands on requirements for system operation and administration to define and administer survivability policies, monitor system usage, respond to intrusions, and evolve system functions as necessary to ensure survivability as usage environments and intrusion patterns change over time.

Evolution Requirements

Lastly, system evolution is an inevitable necessity in response to users' requirements for new functions and intruders' increasing knowledge of system behavior and structure. In particular, survivability requires that system capabilities evolve more rapidly than intruder knowledge. This prevents the accumulation of information about invariant system behavior and structure needed for an intruder to achieve successful penetration and exploitation.

Summary

The emerging discipline of survivable systems is directed at maintaining essential mission operations in adverse circumstances that no amount of security precautions can guarantee to prevent. System survivability can be investigated and improved through the SNA method, and survivability can be integrated into system requirements on a par with functionality and performance. Survivability analysis is a prudent risk management technique in a world of increasing dependency on complex, large-scale network systems.◆

References

1. Lipson, H.F. and Fisher, D.A. *Survivability—A New Technical and Business Perspective on Security,* Proceedings of the New Security Paradigms Workshop, IEEE Computer Society Press, 1999.
2. Presidential Commission on Critical Infrastructure Protection, *Critical Foundations—Protecting America's Infrastructures,* The Report of the Presidential Commission on Critical Infrastructure Protection, October 1997, p. 173., Available at www.pccip.gov
3. DARPA Information Survivability Program. Available at www.darpa.mil/ito/research/is
4. Proceedings of the 1997 Information Survivability Workshop, San Diego, Calif., Feb. 12–13, 1997, SEI and IEEE Computer Society, April 1997. Available at www.cert.org/research
5. Proceedings of the 1998 Information Survivability Workshop, Orlando, Fla., Oct. 28–30, 1998, SEI and IEEE Computer Society, 1998. Available at www.cert.org/research
6. Ellison, R.J., Linger, R.C., Longstaff, T., and Mead, N.R. Survivable Network Systems Analysis: A Case Study, *IEEE Software,* July/August 1999, pp. 70-77.
7. Ellison, R.J., Fisher, D.A., Linger, R.C., Lipson, H.F., Longstaff, T.A., and Mead, N.R. Survivability: Protecting Your Critical Systems, *IEEE Internet Computing,* November/ December 1999.
8. Fisher, D.A. and Lipson, H.F. *Emergent Algorithms—A New Method for Enhancing Survivability in Unbounded Systems,* Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, Maui, Hawaii, Jan. 5–-8, 1999 (HICSS-32), IEEE Computer Society, 1999.
9. Linger, R.C., Mead, N.R, and Lipson, H.F. *Requirements Definition for Survivable Network Systems,* Proceedings of International Conference on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, Calif.,1998, pp. 14-23.

Continued on page 25 ☞

Continued from page 6, **About the Authors**

Ron Knode Ron Knode is the director for Information Assurance Solutions for Computer Sciences Corp (CSC). He leads an operation of nearly 350 information security engineers who serve both commercial and government clients with customized Information Risk Management Program solutions. He also supervises CSC's certified Trust Technology Assessment Program evaluation lab under the National Information Assurance Partnership. He is the author of more than a dozen articles on secure distributed information system architectures and multilevel database management systems.

Ronald B. Knode
Director, Information Assurance, CSC
7459A Candlewood Road
Hanover, Md. 21076
Voice: 410-691-6590
Fax: 410.684.2077
E-mail: rknode@csc.com

Mary Schanken Mary Schanken is a Senior Computer Scientist with NSA who contributed to the development of numerous security documents. She served as government lead for the development of the SSE-CMM, and is implementing the SSE-CMM within the DoD. She is a Lead Assessor and Proficiency Test Grader for laboratories performing Common Criteria evaluations in the United States. She completed her Computer Science degree from the UMBC, and graduate studies at the UMUC, and the Naval War College.

National Security Agency
9800 Savage Rd Ste 6740
Ft. Meade, Md. 20755-6740
Voice: 410 854-4458
Fax: 410 854-6615
E-mail: schanken@nsa.gov

Coming Events 2000

October 15-19
Object Oriented Programming Systems Languages and Applications Conference (OOPSLA 2000)
www.acm.org/events

October 23-25
4th Symposium on Operating Systems Design and Implementation
www.usenix.org/events/osdi2000

October 30-31
3rd International Conference on Practical Aspects of Knowledge Management (PAKM 2000)
www.do.isst.fhg.de/workflow/events/index_e.html

November 10
Infomation Outlook 2000 (Australian Computer Society)
www.acs.org.au/act/events/io2000/index.html

November 16-17
ACM Conference on Universal Usability
www.acm.org/sigchi/cuu

December 4-7
International Conference on Power System Technology
www.ee.uwa.edu.au/~aips/powercon

December 11-13
Global Development Network Conference
www.gdnet.org

April 29-May 3, 2001
Software Technology Conference 2001
www.stc-online.org

The Survivability Imperative: Protecting Critical Systems, continued from page 15, **About the Authors**

Robert Ellison, Ph.D. is a Senior Member of the Technical Staff in the SEI Networked Systems Survivability Program. His research interests include system survivability and architectural patterns and styles for security architectures. He has a doctorate in Mathematics from Purdue University and is a member of the ACM and IEEE Computer Society.

Nancy Mead, Ph.D. is senior member of the technical staff in the SEI Networked Survivable Systems Program, and a faculty member in software engineering, Carnegie Mellon University. She is involved in the study of survivable systems requirements and architectures, and the development of professional infrastructure for software engineers. Prior to joining the SEI, Mead was a senior technical staff member at IBM Federal Systems, where she spent most of her career in development and management of large real-time systems. She also worked in IBM's software engineering technology area, and managed IBM Federal Systems' software engineering education department.

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213

Dr. Thomas Longstaff is senior member of the technical staff at the Software Engineering Institute and currently manages research and development in Survivable Network Technology for the Networked Systems Survivability Program. He is a member of CERT® Coordination Center and conducts analysis of vulnerability and security incidents and methods for assessing survivability. Previously he was technical director at the Computer Incident Advisory Capability at Lawrence Livermore National Laboratory, Livermore, Calif.

Richard Linger is a senior member of the technical staff at the Software Engineering Institute's CERT® Coordination Center at Carnegie Mellon University. He teaches at the CMU H.J. Heinz School of Public Policy and Management. While at IBM, he founded and managed the IBM Cleanroom Software Technology Center. Linger has published three software engineering textbooks and more than 50 articles. He holds a bachelor's degree in electrical engineering from Duke University. He is member of the IEEE, ACM, the National Software Council, and vice-president of the Center for National Software Studies.