

Survivability through Intrusion-Aware Design

Andrew P. Moore and Robert J. Ellison
CERT® Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
{apm,ellison}@cert.org

1. Introduction

Engineers have long relied on public feedback of engineering failures to improve their designs. Imagine what would result if bridge builders had ignored the lessons learned from the torsional oscillations that caused the Tacoma Narrows Bridge to collapse. Or if ship builders had ignored the lessons learned about inadequate lifeboat space and manning that allowed the great loss of life when the Titanic sank. Engineering success requires that we also learn from the less famous disasters. The aerospace community, for example, has institutionalized a means for learning from air traffic accidents that has resulted in very low risk of death during air travel, despite its inherent hazards.

Despite organizational reticence to disclose attacks on their systems, security-relevant failure data have become more available over the past decade. Increased public interest and media coverage of the Internet's security have resulted in increased publication of attack and vulnerability data in books, Internet newsgroups, and vendor/CERT security advisories. Nevertheless, anecdotal evidence indicates that information system engineers are not learning from these documented attacks. Information systems being built, operated, and maintained today are often vulnerable to the same or similar flaws that they have been vulnerable to for years [2].

There are many reasons why the survivability of our systems is not significantly improving. Primary among these is the lack of practical methods for using lessons learned from historical attack data to construct more survivable business processes and information systems. This is consistent with previous observation [1], but little real progress has been made since then. Fortunately, research is starting to focus on this problem [3][7][9][10]. In addition, we have obtained practical results through application of the Survivable Network Analysis (SNA) method, developed at the SEI [4]. SNA uses *intrusion scenarios* – i.e., a description of people and systems interacting in a way that characterizes malicious behavior causing harm to an organization – to improve the survivability of system designs.

This paper describes some relevant insights gained from applying SNA to several significant real-world systems. These insights help understand what is needed to use intrusion scenarios for survivability engineering as part of an *intrusion-aware* development process. Specifically, we describe an organization technique for intrusion scenarios that ameliorates some of the problems that we've encountered. This technique is described more fully in a SEI report [5]. We conclude with several key obstacles to further progress on using intrusion scenarios to improve survivability engineering.

2. SNA Insights

The following describes a few observations of the SNA method used to specify and analyze intrusion scenarios for survivable system analysis. These observations arose as a result of SNA experiences and through detailed discussion with the SNA team members.

® “CERT” and “CERT Coordination Center” are registered in the U.S. Patent and Trademark Office.

Methods must support documenting many diverse intrusions – The ever-increasing number of reported system vulnerabilities and exploits of these vulnerabilities argue for the need to be able to abstract, structure, and organize intrusion scenarios in a usable manner. Evidence indicates that often-ignored social engineering and physical attacks need to be taken at least as seriously as technological attacks [1][8]. Therefore, developers must consider the potential attacks on the whole *enterprise*, not just attacks on its information systems. In addition, attacks by individuals more sophisticated than the average recreational hacker, e.g., industrial spies and international cyber-terrorists, are becoming more likely and more difficult to counter.

Methods must help prioritize intrusion scenarios – Developers cannot afford to defend against all possible attacks. Prioritizing intrusion scenarios according to their likelihood of occurrence and the impact to the enterprise mission is, therefore, critical. This requires understanding the likely adversaries of the enterprise in terms of their capabilities, resources, motivation, risk tolerance, and level of access. Only through this understanding can we provide the optimal defense and recovery of the enterprise mission at an affordable cost.

Methods must work in the face of changing intrusions – A vulnerability can progress through a number of states during its lifetime: birth, discovery, disclosure, correction, publicity, scripting, and death [2]. The state of a vulnerability impacts the importance, or even existence, of intrusions that rely on its exploit. Intrusion scenario documentation and analysis methods must be sensitive to the volatile nature of vulnerability discovery, exploit scripting, and system patching.

Methods must help improve design – The purpose of studying intrusion scenarios is to better understand how to defend against and recover from malicious attacks that could compromise an enterprise's mission. Such improved understanding should lead to improved design, and a more survivable mission operation. Without such an improved understanding, there is little benefit to studying intrusion scenarios.

3. A Useful Organization

Attack trees are a useful way to organize intrusion scenarios. They have existed in various forms, and under various names, throughout the years, but their most recently published form describes them as a systematic method to characterize system security based on varying attacks [8]. Attack trees can refine information about enterprise intrusions by identifying a compromise of enterprise security or survivability as the root of the tree. We refine the ways that an attacker can cause this compromise iteratively and incrementally as lower level nodes of the tree. Nodes may be decomposed either as a sequence of attack steps, represented as an AND-decomposition, or as alternative ways of executing the attack, represented as an OR-decomposition. An attack tree represents a set of intrusion scenarios, each of which accomplishes the mission-critical compromise at the root node by different means [5].

We can improve the survivability of an enterprise by asking resistance, recognition, and recovery questions at each of the attack tree nodes [6]. Resistance questions ask “How can we prevent an attacker from successfully traversing this node to execute an intrusion?” Of course, the answer to such questions may not always be a cost-effective or practical solution. Fundamental to the goal of survivability is recognizing when an attack that we cannot effectively resist takes place and executing recovery plans. We thus ask “How can we detect an attacker during an attempted attack or after a successful attack?” and “How can we react to this detection to prevent an intrusion that might compromise the enterprise's survival?”

Resisting an OR-branch of an attack tree resists the intrusion scenarios associated with that branch. Resisting an AND-branch of an attack tree resists all intrusion scenarios associated with the parent node of the branch. This leverage is gained because the attacker must traverse all branches of an AND-decomposition to execute the attack; resisting any one of the AND-branches counters the attack. Notice that the use of redundancy, or defense in depth, for survivability effectively increases the number of AND-decompositions in an attack tree. This increase generally makes it more difficult to execute an intrusion that leads to the survivability compromise specified at the root node. In addition, resisting attack nodes higher up in the attack tree hierarchy results in more effective blockage of the attacker, but also potentially more extensive and costly changes to the enterprise architecture and operations. The best combination of techniques for resistance, recognition, and recovery is chosen based on cost, practicality, and assurance of implementation.

The practical use of attack trees to document intrusions that compromise real-world enterprise survivability depends on being able to reuse previously developed patterns of attack. We define an *attack pattern* as a generic representation of a deliberate, malicious attack that commonly occurs in a specific context. We organize attack patterns into an encompassing *attack profile* with a common architectural reference model. Different attack profiles may address different levels of attacker access, resources, and skills, as well as different configurations of enterprise components. Therefore, different attack profiles may help refine an application-specific attack tree along different lines of attack [5].

4. Conclusions

Survivability strategies must be integrated with the enterprise operations and design in a spiral-type development process. Considering the possible avenues of attack during this process, i.e., making the process intrusion-aware, is critical to ensuring sufficient protection against and recovery from malicious attack. Attack trees organize related intrusion scenarios in a compact way that relates back to the survivability of the enterprise mission. Attack trees allow the refinement of attacks to a level of detail chosen by the developer. The developer is free to explore certain attack paths in more depth than others while still being able to generate intrusion scenarios that make sense. Refining the leaves of the attack tree simply generates new leaves resulting in intrusion scenarios at the new lower level of abstraction. An attack pattern provides a structure to encode expert security knowledge for reuse. In addition, asking resistance, recognition, and recovery questions at attack tree nodes may suggest improvements to both requirements and design.

Attack trees provide a powerful mechanism to document the multitude of diverse types of attacks on the whole enterprise, to abstract from intrusion details as a buffer against attack volatility, and to suggest improvements to requirements and design. They are, however, only a small part of the answer as to how to use intrusion scenarios to improve survivability engineering. The lack of accurate adversary models and risk analysis methods are serious impediments. A rich attack pattern library populated with attack patterns at the right level of abstraction is needed to build enterprise attack trees more systematically. Finally, the lack of robust resistance, recognition, and recovery countermeasures hampers our ability to improve designs. Overcoming these impediments will require a truly inter-disciplinary effort.

References

- [1] Anderson, R. Why cryptosystems fail. in *Proceedings of the 1st Conference on Computer and Communications Security*, 1993.

- [2] Arbaugh, W.A., W.L. Fithen, and J. McHugh. Windows of vulnerability: a case study analysis. *IEEE Computer*, Vol. 33, No. 12, Dec. 2000.
- [3] McDermott, J. and C. Fox. Using abuse case models for security. in *Proceedings of the 15th Annual Computer Security Applications Conference*, 1998.
- [4] Mead, N.R., R.J. Ellison, R.C. Linger, T. Longstaff, and J. McHugh. Survivable network analysis method. Software Engineering Institute Technical Report CMU/SEI-2000-TR-013, September 2000.
- [5] Moore, A.P., R.J. Ellison, and R.C. Linger. Attack modeling for information security and survivability. Software Engineering Institute Technical Report CMU/SEI-2001-TN-001, March 2001. <<http://www.cert.org/archive/pdf/01tn001.pdf>>
- [6] Moore, A.P., R.J. Ellison, R.C. Linger, and N.R. Mead. Attack modeling for survivable system analysis. in *Supplement of the 2001 International Conference on Dependable Systems and Networks*, Gothenborg, Sweden, 1-4 July 2001
- [7] Salter, C., O. Saydjari, B. Schneier, and J. Walner. Toward a secure system engineering methodology. in *Proceedings of the security Paradigms Workshop*, September 1998.
- [8] Schneier, B. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.
- [9] Tidwell, T., R. Larson, K. Fitch, and J. Hale. Modeling Internet attacks. in *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, 5-6 June, 2001.
- [10] Wood, B.J., J.F. Bouchard, and D.E. Farrell, Jr. Evaluating the effects of adversary behavior on dependable systems. in *Supplement of the 2001 International Conference on Dependable Systems and Networks*, Gothenborg, Sweden, 1-4 July 2001.