Alternative Security Metrics

Jennifer L. Bayuk, *Member, IEEE*

***Abstract*—Today's security metrics support management practices rather than measure system capability to withstand attacks. This eliminates consideration of security features that are not currently used to manage systems as the basis for security metrics. Rather than judge security metrics by a utility standard with respect to current security management practices, they should instead be appreciated for proposing alternatives ways to identify security attributes that may or may not be of use in designing new security management practices. System capabilities such as adaptation to threat, proactive deterrence, and resilience to attack require system capabilities that may be measured using engineering methods for verification and validation of system function.**

***Index Terms*— computer security, data security, systems engineering, metrics**

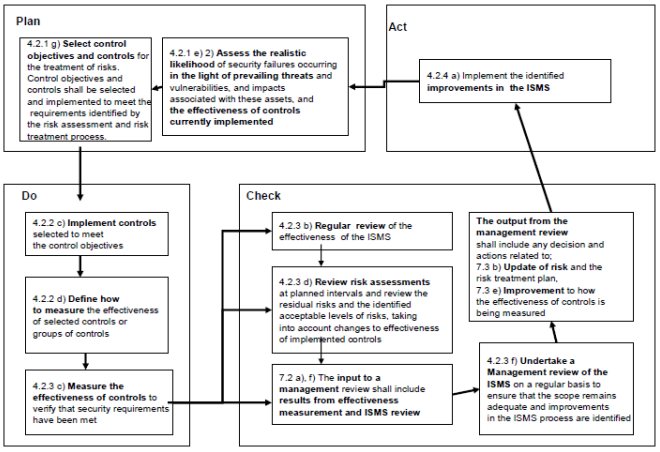
I. Introduction

There are at least 900 measures/metrics that exist in the literature for measuring security because more than 900 are listed in Herrmann's 2007 book, A Complete Guide to Security and Privacy Metrics [1]. Herrmann only included metrics that she considered appropriate for use in decision-making by practicing auditors, engineers, and managers. Herrmann's intent was to create a useful menu for security practitioners, and so she purposely excluded metrics that were abstruse or that relied heavily on an intuitive understanding of complex mathematical models. This idea is echoed in security literature: that metrics form the basis for decisions, and so should be well understood. As Jaquith put it, "transparency facilitates adoption by management" [2 , p.20]. As Pironti put it, "keep it simple" [3].

As Brotby put it, most definitions of security speak to the practice of security, not its objectives [4]. However, Brotby also acknowledges that security that does not lend itself to direct measurement. He refrains from creating a model or a theory of how to directly measure security as a system attribute, but instead defines security in terms of assurance that security goals are met. This leads him to conclude that security metrics should serve the decision-making needs of those whose role it is to provide such assurance; and consequently to conclude that, without management-defined objectives for a security program, it is not possible to develop useful security metrics. Brotby's solution to the security metrics problem thus still leans heavily on the side of technology and operational security metrics from the point of view of the responsible security organization.

Yet, as there is currently no convergence around a single organizational management structure for security, no corresponding authoritative security metrics taxonomy has emerged. However, there has been a great deal of consensus around standards for security management [5-9]. As many security programs are designed to demonstrate compliance with security management standards, they have become defacto metrics taxonomies that cross organizational borders. Practitioners are often advised to organize their metrics around the requirements in security management standards against which they may expect to be audited [1, 2, 10]. As depicted in Figure 1, the International Standards Organization (ISO) even has a standard for using the security management standards to create security metrics [11].

Figure 1: ISO Process for Creating Security Metrics [22]



II. Security Metrics Research

The result of this focus on the practitioner is that current security research in metrics that are not practical given today's security management structures are deemed "not useful." They are excluded from standards documents, and thus also from methods and tools that engineers currently use to determine security requirements. These include security metrics for mathematical modeling of security management processes [12], weighting network forensics evidence to increase probabilities of conviction [13], quantifying threat surface using hidden Markov models [14], using game theory to determine security investment strategies [15], and complex

Manuscript received October 21, 2010. This work was supported in part by the U.S. Department of Defense under contract H98230-08-D-0171.

Jennifer Bayuk is with Stevens Institute of Technology, Hoboken, NJ 07030 USA (973-335-3530; fax: 973-335-0789; e-mail: jennifer.bayuk@stevens.edu).

mathematical models for assessing software security [16]. Most of these are the subject of one or two papers by the same group of authors, and rely on data that is not completely described (and also usually includes subjective measures of probability). A practitioner reading these types of hypothetical usefulness cases recognizes that there are more straightforward ways to assess the security in their target environment, and that is why this set was excluded from Herrmann's compilation, and also the standards literature.

Nevertheless, metrics in the "not useful" category may in fact provide some useful clues as to how we may secure systems in ways that our current management structure and standards do not consider. Rather than judge these research contributions by a utility standard with respect to current security management practices, they should instead be appreciated for proposing alternatives ways to identify security attributes that may or may not be of use in designing new security management practices.

For example, Clark et.al., argued that no amount of close attention to software security was a better approach than simply changing the code on a regular basis [17]. They noted that software is rarely successfully attacked the day it is deployed, not because it is not vulnerable, but because hackers do not yet know how to attack it. Thus, a new software architecture will generally enjoy a "honeymoon period" between the first release of a program and the disclosure of its first vulnerability. In its honeymoon period, software operates unthreatened. This is in contrast to older and more mature software that hackers have had time to examine in depth, and so they will find very obscure bugs in older software even if it has fewer vulnerabilities overall than comparable software that is still in its honeymoon period. Clark observed that, in order to be able to predict when a piece of software is reaching the end of its honeymoon period, we must be able to measure how long it will be before a system is first attacked.

III. A New Approach to Security Metrics

The rise of the self-adapting and mutable botnet has made it clear that hackers have already adopted systems security approaches that take advantage of an ability to change in the face of determined attack [18]. However, as long as we continue to measure security with reference to today's security management practices, we will continue to discard the measure of actual system resilience in the face of a determined and agile adversary.

One outspoken security practitioner, Ed Amoroso, has discussed some of these issues in a book on Cyber Attacks[19]. In it, he presented several new ways of thinking about security that may offer a clue to the future of security metrics. For example, he asked: How might your system employ intentional deception? How might it diversify its threat surface? How might it increase situational awareness? Where answers to these questions are used to measure security, new types of systems features will emerge.

The need for new ways of engineering security was recently addressed in the SERC Systems Security Research Roadmap [20]. In that paper, it was emphasized that the concept of security should allow it to be understood as a tangible systems attribute. Security provides safeguards that contribute to a system's ability to achieve its mission and purpose in the face of changing threats. By this definition, system functions for proactive, adaptive, and resilient behavior in the face of adversaries may be included in system capability requirements. A clear understanding of the definition of security in the context of a given system mission should allow the design of alternative security features, as well as metrics to determine their effectiveness in maintaining system security.

A recent debate among respected security engineers produce a wiade variety of definitions for security. These included:

Bayuk: something that thwarts people (and/or systems acting on their behalf) who, intentionally or not, enact threats that exploit system vulnerabilities and thereby cause damage that adversely impacts system value.

Geer: the absence of unmitigitable surprise

Turner: a form of protection where a separation is created between the assets and the threat. This includes but is not limited to the elimination of either the asset or the threat. In order to be secure, the asset is removed from the threat, or the threat is removed from the asset

Seierson: mitigating all known risks that are worth mitigating and hope that reduces the risk of the unknown risks too. A working definition of the unknown risks would be "all the stuff we don't know about".

Thomas: Security cannot be defined by any specification of the states of the world. Instead, security is a judgment about the present state of the world relative to a future we believe is foreboding. Therefore, any specifications and measurements you make about the state of the world (people, process, and technology) will only become meaningful as security when the judgment is made. "Risk" is the cost of the future(s) brought to the present, and is inextricably linked to judgments about security. In sum, you can't define security without explicitly incorporating the time dimension and the epistemic frame.

Rather than interpret the divergence as a reason for giving up hope, a systems engineer should view these divergent definitions as different stakeholder perspectives. It is common for a systems engineer to confront differing stakeholder perspectives of the same requirements. Security should be no exception.

For example, a requirements set that satisfies the above definitions incorporates industry standard checklists for patches and software hardening techniques known to thwart threats documented in the NIST National Vulnerability Database [1], a threat surface taxonomy [2], an enterprise risk management standard [3], and a common sense test, ad enunciate by Geer, above.

The hope for a more complete understanding of what is meant by security therefore seems to lie in construct validity. Construct validity starts with a theory. One reason for this is

because it is acknowledged that a construct validity test may fail for reasons other than the hypothesis being false. They could fail testing because the test was not adequately designed. The difficulty in designing vulnerability tests has already been identified. The difficulty in designing audit and assessment tests is not as well understood. Using a single audit to test an entire organization's security is at least a 3 step process (and may include more, see [4]). It starts with creating a model of how security to supposed to be accomplished in that organization, as identified by management. It then evaluates whether the model would be effective if all elements described by it were functioning correctly. Finally, it tests key controls identified by the model to see if they are working [for example, see [5].

IV. Security Requirements Convergence

Emerging security architecture frameworks allow a systems engineer to merge stakeholder perspectives into system capabilities that satisfy all perspectives. This methodology may extend and enhance systems architecture to produce security requirements at the system rather than at the security technology level [21]. Of course, due to the possibility of threats that are unknown, no system will ever be 100% secure. Nevertheless, this approach should enable a new type of security metrics by:

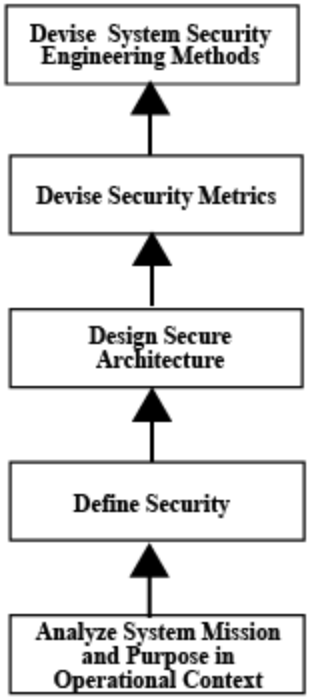
- Identifying security features that require system-level functions.
- Evaluate the extent to which security features protect systems from deliberate damage that would cause system failure.
- Devise verification and validation metrics at the system level that show security requirements are met.

Figure 1 illustrates the recommendation of the roadmap. Security is defined in terms of the mission and purpose of the system of interest, it comes from the context within which a system operates. This view of security as an enabler allows security architecture to be functions of systems architecture, customized rather than bolted-on. Security architecture metrics may then measure whether security functional requirements are met. Where systems exhibit similar architecture patterns, it is expected that they will have similar security architecture requirements. The existence of common security architecture models should make it possible to develop tools that may be developed to guide future engineering efforts toward more secure solutions. A key element of the systems security engineering roadmap is to provide capability for security researchers to self-assess the value of a potential contribution to the field by using security metrics to determine whether a system is better able to perform its mission and achieve its purpose as opposed to measuring whether it allows a security practitioner to more easily manage security features.

V. Conclusion

This paper describes the current state of the practice in security metrics. It provides examples of research in security metrics that may be used to extend the state of the art. It describes research in systems security engineering that provides a framework that has the potential to improve the quality of the state of the practice. It recommends exploration of security metrics that measure a system's ability to withstand attacks.

Figure 2: Security Engineering Methodology



References

[1] *National Vulnerability Database*. Available: http://nvd.nist.gov/

[2] P. Herzog, "Open Source Security Testing Methodology Manual, V.3.2, http://www.isecom.org/osstmm/," 2010.

[3] Information Security Forum, "The Standard of Good Practice for Information Security," ed, 2007.

[4] J. Bayuk, *Stepping Through the IS Audit, A Guide for Information Systems Managers*, 2nd ed.: Information Systems Audit and Control Association, 2005.

[5] American Institute of Certified Public Accountants, "Auditing Practice Release No. 021056: Implementing SAS No.#70 Reports on the Processing of Transactions by Service Organizations.," ed.

Jennifer L. Bayuk (M'08) is the Cybersecurity Program Director of the School of Systems and Enterprises at Stevens Institute of Technology. She develops graduate curriculum for security systems engineering and enterprise security architecture, as well as leading research in systems security engineering. Bayuk has been a Wall Street Chief Information Security Officer, a Manager of Information Systems Internal Audit, a Price Waterhouse Security Principal Consultant and Auditor, and a Security Software Engineer at AT&T Bell Laboratories. Bayuk frequently publishes and speaks on IT Governance, Information Security, and Technology Audit topics.

She is the author of *Stepping Through the IS Audit, 2nd Edition* (ISACA 2004), *Stepping Through the InfoSec Program* (ISACA 2007), and Enterprise Security for the *Executive* (Praeger, 2010). She also has edited a collection of

works on *Cyberforensics* (Springer 2010) and co-edited a collection of works on *Enterprise Information Security and Privacy* (Artech House, 2009).

Ms. Bayuk is a Certified Information Security Manager, a Certified Information Systems Security Professional, a Certified Information Security Auditor, and Certified in the Governance of Enterprise IT (CISM, CISSP, CISA, and CGEIT), as well as a member of IEEE, INCOSE, and ACM. She has Masters Degrees in Computer Science (SIT) and Philosophy (OSU).