Information Security System Rating and Ranking

Dr. Rayford B. Vaughn Jr., Ambareen Sira, and Dr. David A. Dampier
Mississippi State University

Wednesday, 1 May 2002
Track 5: 9:00 - 9:40
Room 250 A - C

The term assurance has been used for decades in trusted system development to express the notion of confidence in the strength of a specific system or system of systems. The unsolved problem that security engineers must struggle with is the adoption of measures or metrics that can reliably depict the assurance associated with a specific hardware and software architecture. This article reports on a recent attempt to focus needs in this area and suggests various categories of information assurance metrics that may be helpful to an organization that is deciding which set is useful for a specific application.[1]

We believe that the provision of security in systems is a subset of the systems engineering discipline, and that it has a heavy software-engineering component. As software engineers, we understand that the determination and application of measures and metrics is not an exact science, nor is it easily accomplished. We also realize that this difficulty carries over to the trusted systems world. How one measures the degree of protection present is, today, an unsolved question and is primarily accomplished by craftsmanship and not science.

This issue of rating and ranking systems in terms of their assurance characteristics was at least partially addressed at a workshop on information security system ratings and ranking in Williamsburg, Va.,[2] in spring 2001. We will hereafter refer to this as *the workshop*, as we reference it in support of our belief.

Workshop Findings and Observations

It appears that while we often claim to have metrics that prove or indicate assurance levels, we do not seem to be able to prove that correctness, maintainability, reliability, and other such nonfunctional system requirements are in the software we build. We also tend to use empirical evidence based on historical performance data in claiming system strength. However, knowing that a particular defensive strategy has worked well in the past for an organization really says very little about its strength for the future. Examples of software engineering difficulties that we face in predicting a system's strength include the following:

- Software is not subject to the laws of physics. In most cases, we cannot apply mathematics to code to prove correctness in the same way a bridge builder can apply formulae to prove structural strength characteristics.
- People who are, by nature, error prone build software. In the end, any one of them can intentionally or unintentionally corrupt the system and greatly diminish assurance.
- Compositions of mechanisms used to construct a security perimeter comply with no known algebra. We remain reliant on the expertise of our systems administrators or security engineers.

> *"... while we often claim to have metrics that prove or indicate assurance levels, we do not seem to be able to prove that correctness, maintainability, reliability, and other such nonfunctional system requirements are in the software we build."*

- It is easier to attack a system today (an assurance issue) than it was years ago. This trend is likely to continue as attack tools are further automated, shared, and explored on a global basis.

The workshop attempted to address these issues and others. Although many specific techniques and suggestions were proffered to the group, it was apparent to all that some combination of measures was essential, and that this combination could not generically be applied across all interest domains. Similarly, it was clear that the measures or metrics adopted by an organization to determine assurance need to be frequently revisited and re-validated.

Attempts to apply a single rating to a system have been tried in the past and have failed [1, 2]. The workshop organizers also agreed that the problem domain might be best viewed using a non-disjoint partitioning into technical, organizational, and operational categories.

Definitions agreed upon by the conference organizers in the technical category were measures/metrics that are used to describe and/or compare technical objects (e.g., algorithms, products, or designs). Organizational measures might be used with respect to processes and programs. Operational measures are thought to describe *as is* systems, operating practices, and specific environments.

An interesting characterization of information security metrics came from Deb Bodeau of The Mitre Corporation [3] who pointed out that a proper view of these metrics might be a cross product involving what needs to be measured, why you need to measure it, and for whom you are measuring. Her characterization of this view in Figure 1 is enlightening.

Another interesting observation made by several attendees was that the desired purpose for such measures and metrics seemed to vary between the government and commercial sectors. Government applications seem much more likely to use metrics and measures for upward reporting. Answering such questions as "What is our current assurance posture?" "How are we doing this month compared with last?" and "Are we compliant with applicable regulations and directives?" seemed to be a driver for the metrics needed by government.

The representatives at the workshop from the commercial world seemed less interested in these questions and more inclined to look for answers to the ques-

tions "How strong is my security perimeter?" "What is the return on my investment?" "What is my level of risk or exposure?" and "How does product performance compare?" The commercial sector seemed to have far more interest in technical and operational measures than in process or organizational measures.

The workshop attendees had hoped to find a number of objective, quantitative metrics that could be applied. Although unanimous agreement was not reached, it was apparent to most that such metrics were in short supply, had to be combined with other measures or metrics in a particular context, and were generally not very useful on their own. Many more measures that would be considered subjective and/or qualitative appeared more useful.

Examples of more useful measures might include adversary work factor – a form of penetration testing. An excellent discussion of this topic is found in Schuedel and Wood [4]. Although penetration techniques are not truly repeatable and consistent, the workshop attendees agreed that their results were meaningful and useful. Risk assessments, in their various forms, were also found to be useful measures of assurance. Such assessments are accomplished in a variety of ways, but tend to focus attention in the proper areas and give a good indication of how one is postured to withstand attacks on a system.

Information Assurance (IA) metrics are essential for measuring the *goodness* of IA, and we believe that overall useful IA metrics are possible. There is general agreement that no single system metric or any *one perfect* set of IA metrics applies across all systems. Which set will be most useful for an organization largely depends on its IA goals; its technical, organizational, and operational needs; and the resources that it can make available.

In order to help an organization investigate options for IA metrics, it is useful to look at the different categories of IA metrics in general. These categories are described as follows:[3]

Objective/Subjective

Objective IA metrics (e.g., mean annual downtime for a system) are more desirable than subjective IA metrics (e.g., amount of training a user needs to have to securely use the system). Since subjectivity is inherent in IA, subjective IA metrics are more available.

Quantitative/Qualitative

Quantitative IA metrics (e.g., number of failed login attempts) are more preferable than qualitative IA metrics (e.g., the Federal Information Technology Security Assessment Framework [5] self-assessment levels).

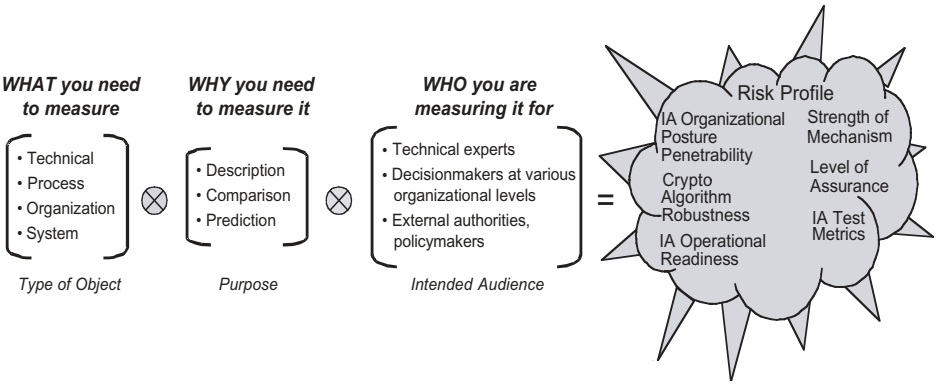

Figure 1: *Characterization of Information Security Metrics (Bodeau)*

Static/Dynamic

Dynamic IA metrics evolve with time, static metrics do not. An example of static IA metrics can be the percentage of staff who received annual security refresher training [3]. This metric can degrade in value if the content of the course does not change over time. An example of dynamic IA metrics can be the percentage of staff who received training on the current version of the software package they use. Most metrics used in penetration testing are dynamic. Dynamic IA metrics are more useful than static because best practices change over time with technology. There is always need to adapt metrics in compliance with best practices [6].

Absolute/Relative

Absolute metrics do not depend on any other measures, and these either do or do not exist [3]. For example, the number of systems administrator, networking, and security-certified security engineers in an organization is an absolute metric. Relative metrics are only meaningful in context. For example, the number of vulnerabilities in a system cannot assess the system's IA posture alone. The type and strength of vulnerabilities are also important in this context for making any decision about the system's IA posture. The majority of IA metrics are relative, and so they would not be good for use as a single-system metric.

Direct/Indirect

Direct IA metrics can be generated from observing the property that they measure. For example, the number of invalid packets rejected by a firewall over a certain period of time. Indirect IA metrics are derived by evaluation (e.g., ISO Standard 15408 The Common Criteria) and/or assessment (e.g., risk assessment). Although preferred, sometimes it is not possible to measure directly. In these cases, indirect measures are useful.

IA is a triad of cooperation between the technology that provides assurance, the processes that leverage that technology, and the people who apply and make the technology work [7]. IA metrics should be all encompassing – the product, the process, and the people, because processes build products that people use. If we want to be assured that proper information protection is in place, we need to know what it is that we wish to protect, that we have the right product for protection, that the product was built correctly, and that the right people are using it properly.

Summary

The workshop was successful in focusing attention on the area of metrics or measures for systems that have security or assurance as a requirement. It was not successful in getting agreement on a set of measures to be used, or even finding consensus in any particular approach. Nonetheless, several themes emerged from this workshop that may be useful. These are reported below, as taken from the draft proceedings of the workshop at the time of this writing.

- There will be no successful single measure or metric that can quantify the assurance present in a system. Multiple measures will most certainly be needed, and they will need to be refreshed frequently.
- Software and systems engineering are very much related to this problem: The quality of the software delivered, the architectures and designs chosen, the tools used to build systems, the specified requirements, and more are all related to assurance.
- Penetration testing is, today, a valid measurement method. It is imperfect and to some extent non-repeatable, but nonetheless, it is used in both government and the commercial sectors. Several other testing measures are valuable: They include level of effort, numbers of vulnerabilities found (or not

found), and number of penetrations.
- There are differences between the government and the commercial sectors. One is policy driven – the other is profit driven. Defense in depth and breadth is important. Knowing how to measure this defense is also important and a valid research area. There was no agreement on how to accomplish this measurement.
- Attempts to quantify and obtain a partial ordering of the security attributes of systems in the past have not been successful to a large degree (e.g., the Trusted Computer Systems Evaluation Criteria and the Common Criteria [1, 2]).
- Processes, procedures, tools, and people all interact to produce assurance in systems. Measures that incorporate all of these are important. We believe Bodeau has characterized this very well in Figure 1 (see page 31).

References

1. Department of Defense Standard. Department of Defense Trusted Computer System Evaluation Criteria. DOD 5200.28-STD, GPO 1986-623-963, 1985.
2. ISO Standard 15408. The Common Criteria.
3. Workshop on Information-Security-System Rating and Ranking, Williamsburg, Va. 21-23 May 2001. Draft Proceedings (unpublished). <www.acsac.org/measurement/position-papers/index.html>.
4. Schuedel, G., and B. Wood. "Adversary Work Factor as a Metric for Information Assurance." Proceedings of the New Security Paradigm Workshop, ACM/SIGSAC, Ballycotton, Ireland 18-22 Sept. 2000. <wnspw.org> (ACM order number 537001).
5. National Institute of Standards and Technology, Computer Security Division, Systems and Network Security Group. Federal Information Technology Security Assessment Framework. NIST, 2000. <http://csrc.nist.gov/organizations/guidance/framework final.pdf>.
6. Bartol, N. "IA Metrics Development and Implementation." In a position paper submitted to the workshop on Information-Security-System Rating and Ranking, Williamsburg, Va. 21-23 May 2001. Booz-Allen & Hamilton, 2001.
7. McCallam, D. "The Case Against Numerical Measures of Information Assurance." In a position paper submitted to the Workshop on Information-Security-System Rating and Ranking, Williamsburg, Va. 21-23 May 2001. Logicon, 2001.

Notes

1. This work is partially sponsored by the National Science Foundation Grants CCR-0085749 and CCR-9988524.
2. Sponsored by the MITRE Corporation and the Applied Computer Security Associates.
3. The categories outlined here are from research at Mississippi State University's Center for Computer Security Research, <www.cs.msstate.edu/~security>, in a larger effort to create a taxonomy for information assurance metrics and measures. This work can be shared by contacting the authors of this article.

About the Authors

Rayford B. Vaughn Jr., Ph.D., is professor of computer science at Mississippi State University. A retired Army Colonel, he served 26 years, including commanding the Army's largest software development organization and creating the Pentagon Single Agency Manager organization to centrally manage all Pentagon information technology support. After retiring from the Army, he was vice president of Integration Services, Electronic Data Systems Government Systems. Dr. Vaughn has more than 40 publications and actively contributes to software engineering and information security conferences and journals. He holds a doctorate degree in computer science from Kansas State University.

Department of Computer Science
P.O. Box 9637
Mississippi State University
Mississippi State, MS 39762
Phone: (662) 325-2756
Fax: (662) 325-8997
E-mail: vaughn@cs.msstate.edu

Ambareen Siraj is a graduate student in the Computer Science Department at Mississippi State University and a member of the Computer Security Research Center. She is working toward a doctorate degree under the direction of Dr. Rayford B. Vaughn Jr. Her research combines the use of artificial intelligence techniques in the creation of a network-based decision engine designed to fuse and analyze information from multiple intrusion detection sensors. She also has a strong interest in the area of measuring the trustworthiness of systems and in the use of metrics and measures to do so. She has written several papers on her work and continues to be active in that endeavor.

Department of Computer Science
P.O. Box 9637
Mississippi State University
Mississippi State, MS 39762
Phone: (662) 325-2756
Fax: (662) 325-8997
E-mail: ambareen@cs.msstate.edu

David A. Dampier, Ph.D., served over 20 years in the U.S. Army, the last 12 as a software engineer and automation officer. In that capacity, Dr. Dampier conducted research in software prototyping and software evolution at the Army Research Laboratory, and he taught software and information engineering at the National Defense University. In February 2000, he left the Army to join the computer science department at Mississippi State University where he teaches software engineering and computer science. Dr. Dampier's research interests are in formal methods for software engineering and software evolution, software process automation, and computer forensics.

Department of Computer Science
P.O. Box 9637
Mississippi State University
Mississippi State, MS 39762
Phone: (662) 325-8923
Fax: (662) 325-8997
E-mail: dampier@cs.msstate.edu