To appear IEEE HICSS-39 Poipu Kauai, Hawaii January 4-7, 2006 (http://www.hicss.hawaii.edu/).

Challenges in Cyber Security

Ali Mili,
College of Computing Science
New Jersey Institute of Technology
Newark NJ 07102-1982
(973)596 5215, mili@cis.njit.edu

Frederick Sheldon
U.S. DOE Oak Ridge National Lab
PO Box 2008, MS 6085
Oak Ridge TN 37831-6085
(865)576 1339, sheldonft@ornl.gov

1 The Security Gap: A Looming Crisis

The term *missile gap* was coined in the late fifties when the Soviets launched Sputnik, to refer to the wide gap between the US's national capabilities in space and its national aspirations. The term *software gap* was coined by Dr Harlan D. Mills in the mid eighties, at the height of the IT revolution, when it was clear that software technology was not keeping up with the demands of the national economy as it was growing increasingly dependent on the safe, reliable operation of software systems.

We coin the term *Security Gap* to refer to the vast technological gap that exists today between available capabilities and the demands imposed by recent global developments. We submit that the security gap matches or exceeds the earlier gaps in terms of its stake, and in terms of its technical challenge. We further argue that not only is the security gap very wide, it is actually widening, making it even more urgent that we devote our attention to this technology. Figure 1shows the three dimensions of security:

- *Criticality.* There is no need for security unless we have a critical resource to protect.
- *Threat.* There is no need for security unless there is a threat to acquire, to expose, or to damage the critical resource that we have.
- *Vulnerability.* Even in the presence of criticality and threat, there is no need for security unless the system that protects the critical resource is vulnerable, in the sense that it lends itself to attack/ intrusion/ exposure/ damage, etc.

We claim that the security gap is widening nowadays, as we are faring worse on all 3 dimensions of security:

- We have increased threat, as a consequence of the emerging global tensions in the world today, as well as the increased sophistication of perpetrators.
- We have increased criticality because the emergence of the Internet has shifted more and more economic and social activity online, making security virtually synonymous with cyber-security.
- We have increased vulnerability because emerging computing paradigms, such as networking, distributed computing, mobile computing, open wide security gaps that are hard to control.

In this short position paper, we argue that a quantitative, formal approach is needed for modeling system security, and briefly discuss the outline of a refinement based approach that integrates security with other dimensions of dependability.

2 Managing Security: A Quantitative Approach

It is fair to characterize current security management techniques by means of two broad premises:

- *They are primarily Defensive.* Traditional security measures are primarily defensive, in the sense that they are geared to predefined, pre-cataloged, vulnerabilities and threats, and are unable to deal with unknown threats.
- *They are primarily Qualitative.* Security measures are taken because they are perceived to help with predefined threats; there is no quantification of how well they deal with the specific threat they are intended to mitigate, nor with any other variations on this threat.

This manuscript has been authored by a contractor of the U.S. Government (USG) under DOE Contract # DE-AC05-00OR22725. The USG retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for USG Purposes.

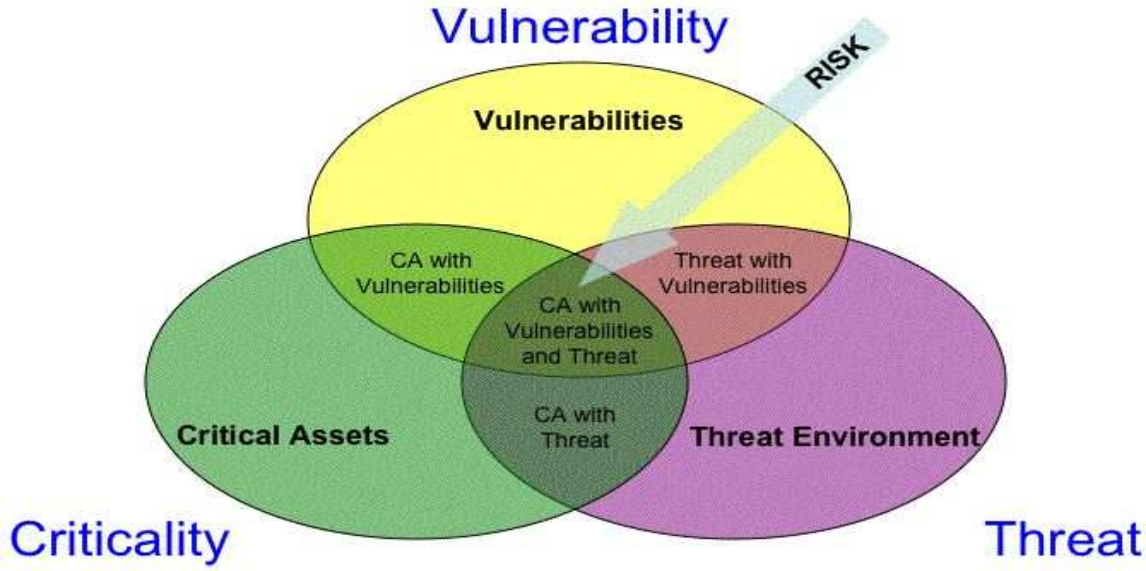

Figure 1: Three Dimensions of Security

- *They are primarily Ad-Hoc.* When security requirements are perceived to be high (due to high criticality, high vulnerability, or high level of threat), several measures are taken in concert; there is no acknowledgment of whether these measures are redundant or complementary, whether they overlap, whether together they ensure that some security goal is met, etc.

It is necessary to bring some discipline to security management, by providing the following elements: security management, by providing the following elements:

- A logic for specifying security requirements and verifying secure systems against such requirements.
- A model for managing system security by quantifying costs, risks, measures and countermeasures.
- Automated tools that support security management according to the proposed models.

3 Modeling Security

Even though logically, system reliability is driven exclusively by the existence and possible manifestation of faults, empirical observations regularly show a very weak correlation between faults and reliability. In [13], Mills and Dyer discuss an example where they find a variance of 1 to 50 in the impact of faults on reliability; i.e. some faults cause system failure 50 times more often than others; while their experiment highlights a variance of 1 to 50, we have no doubt that actual variance is in fact unbounded. Also, they find that they can remove 60 percent of a system's faults and improve its reliability by only ... 3 percent. In a study of IBM software products, Adams [1] finds that many faults in the system are only likely to cause failure after hundreds of thousands of months of product usage.

We argue that the same may be true for security: vulnerabilities in a system may have widely varying impacts on system security. In fairness, the variance may be wider for reliability than for security, because in malicious security violations high impact vulnerabilities may be more attractive targets than lower impact vulnerabilities, but wide variances are still quite plausible. Wide variances, to the extent that they are borne out, have broad impacts on security management:

- In practice, security ought not be defined as the absence of vulnerabilities, no more than reliability is defined by the absence of faults (low impact vulnerabilities do not affect security in a meaningful way).

- In practice, security ought not be measured or quantified by the number of vulnerabilities, just as it is widely agreed (as highlighted by Adams' [1] and Mills' [13] work) that faults per KLOC is an inappropriate measure of reliability.

- Security cannot be improved by focusing on vulnerabilities, as we have no way to tell whether a given vulnerability has low (1) or high (50) impact on security. Rather, security should be managed by pursuing a policy that leads us to the highest impact vulnerabilities first (a similar approach to usage pattern testing [3, 5, 9–11, 13, 14, 16]).

In light of these observations, we argue in favor of modeling security in a way that reflects its visible, measurable, observable attributes, rather than its hypothesized causes. We argue that a *Logic for System Security* is needed, to be defined in terms of the following features:

- A notation for *security specification*, which details how to capture security requirements of a system.

- A formula for *security certification*, which formulates the condition under which a system (represented by its security abstraction) meets a given set of security requirements (represented by security specifications).

Note that in order to quantify reliability as the mean time to *failure*, we must define what it means to *fail*, which in turn requires that we define *specification* and *correctness*. Likewise, defining and quantifying security requires that we define the concepts of security specification and security certification.

4 Security as a Dimension of Dependability

It is customary [17] to define *dependability* as the aggregate of four attributes: *availability* (probability of providing services when needed), *reliability* (probability of failure free operation), *safety* (probability of disaster free operation) and *security* (probability of interference free operation). An important distinction is between *availability*, which deals with operational properties, and the other three, which deal with functional and behavioral properties. We argue in favor of a uniform model to capture the three behavioral properties, i.e. reliability, safety and security. We submit three broad arguments to support our position.

- *Conceptual Argument.* Generalization is a problem-solving strategy that substitutes a specific problem with a more general problem, thereby abstracting away many irrelevant problem-specific details; though it is paradoxical, it is an effective problem solving strategy, as it tends to abstract away irrelevant detail, whence produce more elegant solutions. We argue that reliability, safety and security lend themselves to analysis by generalization, by virtue of their commonalities. Reliability is defined in terms of faults, errors and failures, and is handled by means of a hierarchy of three methods: fault avoidance, fault removal and fault tolerance. Safety is defined in terms of three concepts, hazard, mishap and accident, and is handled by means of a hierarchy of three methods: hazard avoidance, hazard removal and damage limitation. Security is defined in terms of three concepts, vulnerability, threat and exposure (or attack), and is handled by means of a hierarchy of three methods: vulnerability avoidance, vulnerability neutralization, and exposure limitation. We argue that these analogies are a strong hint to at least attempt to model these attributes in a uniform manner.

- *Pragmatic Argument.* Reliability, safety and security are interdependent, in the sense that each property may depend on the others to hold. For example, all the claims of reliability and safety become void if an intrusion occurs and alters the system's function or the system's state. Conversely, the security of a system is dependent on the reliability of the components that implement/ enforce its security measures. Hence in practice, having high values for one of these attributes is probably not meaningful unless we have commensurate values for the other attributes as well; also, it is conceivable that the proof of any one of these properties will use hypotheses about the other properties. In addition, while the distinctions between reliability, safety and security are meaningful for the engineer, they are less meaningful for the user: From the standpoint of the user, it matters little whether a system failed because of violation of a reliability requirement, a safety requirement,

or a security requirement.

- *Methodological Argument.* By virtue of the analogies that we highlighted above, it is very likely that methods developed for one property prove to be useful to other properties. In particular, it is quite plausible that methods developed for ensuring reliability prove to be useful in ensuring security.

5 A Refinement Based Approach

Whereas in the previous subsection we argued in favor of a uniform approach to the three behavioral attributes of dependability (reliability, safety, security) in this subsection we argue in favor of a specific uniform approach, one based on refinement calculi. We submit three broad arguments to support our position.

- *Conceptual Argument.* Refinement calculi have long been used to model correctness properties, hence form the basis for reliability analysis [2, 4, 6–8, 18]. In [12] Mili et al have shown how safety can be modeled by the same refinement mathematics, and how reliability and safety concerns can be addressed with the same recovery mechanism. We submit that the same refinement mathematics can be used to model (some aspects of) security. In particular, we argue that there is no difference between reliability and safety except in the quantification of failure costs (the violation of a safety requirement costs more than the violation of a reliability requirement). We also argue that there is no difference between reliability and security except in the characterization of fault hypotheses (reliability deals with hardware or software faults while security deals with faults caused by malicious intervention).

- *Pragmatic Argument.* In [15], Nicol et al submit that it is virtually impossible to ascertain the security of a complex system, and argue that in order to enhance system security we must deploy a wide range of methods. We argue that many refinement based methods of reliability are designed to achieve this very same goal; they can be generalized or adapted to deal with security. An important attribute of these methods is their ability to decompose verification goals and to compose verification claims, as a *divide-and-conquer* discipline.

- *Methodological Argument.* By merging various aspects of dependability and modeling them in a uniform manner we make it possible to build eclectic arguments of dependability.

References

[1] E.N. Adams. Optimizing preventive service of software products. *IBM Journal of Research and Development*, 28(1):2–14, 1984.

[2] R.J. Back and J. von Wright. *Refinement Calculus: A Systematic Introduction.* Graduate Texts in Computer Science. Springer Verlag, 1998.

[3] S.A. Becker and J.A. Whittaker. *Cleanroom Software Engineering Practice.* IDEA publishing, 1997.

[4] Jules Desharnais, Ali Mili, and Thanh Tung Nguyen. Refinement and demonic semantics. In Ch. Brink, W. Kahl, and G. Schmidt, editors, *Relational Methods in Computer Science*, chapter 11, pages 166–183. Springer Verlag, January 1997.

[5] M. Dyer. *The Cleanroom Approach to Quality Software Development.* John Wiley and Sons, Inc., 1992.

[6] P. Gardiner and C.C. Morgan. Data refinement of predicate transformers. *Theoretical Computer Science*, 87:143–162, 1991.

[7] E.C.R. Hehner. *A Practical Theory of Programming.* Springer-Verlag, 1993.

[8] M.B. Josephs. An introduction to the theory of specification and refinement. Technical Report RC 12993, IBM Corporation, July 1987.

[9] R.C. Linger. Cleanroom software engineering for zero-defect software. In *Proceedings, 15th Hawaii International Conference on Software Engineering*, Baltimore, MD, May 1993.

[10] R.C. Linger. Cleanroom process model. *IEEE Software*, 11(2):50–58, 1994.

[11] R.C. Linger and P.A. Hausler. Cleanroom software engineering. In *Proceedings, 25th Hawaii International Conference on System Sciences*, Kauai, Hawaii, January 1992.

[12] Ali Mili, Frederick Sheldon, Fatma Mili, and Jules Desharnais. Recoverability preservation: A measure of last resort. *Innovations in Systems and Software Engineering: A NASA Journal*, 2005.

[13] H.D. Mills and M. Dyer et al. Cleanroom software engineering. *IEEE Software*, 4(5):19–25, 1987.

[14] H.D. Mills, R.C. Linger, and A.R. Hevner. *Principles of Information Systems Analysis and Design*. Academic Press, 1985.

[15] David M. Nicol, William H. Sanders, and Kishor S. Trivedi. Model based evaluation: From dependability to security. *IEEE Transactions on Dependable Computing*, 1(1):48–65, 2004.

[16] S.J. Prowell, C.J. Trammell, R.C. Linger, and J.H. Poore. *Cleanroom Software Engineering: Technology and Process*. SEI Series in Software Engineering. Addison Wesley, 1999.

[17] Ian Sommerville. *Software Engineering*. Addison Wesley, seventh edition edition, 2004.

[18] J. Von Wright. A lattice theoretical basis for program refinement. Technical report, Dept. of Computer Science, Åbo Akademi, Finland, 1990.