Science of Security Hard Problems: A Lablet Perspective

David M. Nicol (Univ. of Illinois at Urbana-Champaign)
William H. Sanders (Univ. of Illinois at Urbana-Champaign)
William L. Scherlis (Carnegie Mellon Univ.)
Laurie A. Williams (North Carolina State Univ.)

November 27, 2012

Introduction

The development and operation of secure systems continue to present significant technical challenges. Considerable research effort is underway in the technical community to address these challenges. A vast literature of technical results demonstrates progress, with tangible benefit to practice. But many of the most fundamental technical problems remain open, creating fundamental difficulties in the engineering, evaluation, and operation of systems intended to be secure. To assist in addressing these challenges, the U.S. National Security Agency initiated a coordinated set of focused research activities undertaken under the auspices of three Science of Security Lablets, which are sited at the University of Illinois at Urbana-Champaign, the North Carolina State University, and Carnegie Mellon University.

The Lablets share a broad common goal, which is to develop the foundations for security science, with a focus on advancing solutions to a selection of the hardest technical problems. The goal is to develop foundations for the *science of security*—an explicit intellectual framework of process and methods for advancing scientific understanding and accelerating the transition of scientific results into practice. Because our investigations build on both mathematical and empirical approaches, our framework of scientific advancement must focus on both abstract models and data—and on their interplay as a source of validation and modeling capability. Based upon mathematical or empirically-based reasoning, abstract models capture selected salient features of the world for the purpose of prediction. Sound science leads to sound predictions. Sound predictions have fewer (or more manageable) caveats due to modeling choices and have greater operational validity. The predictions can pertain to the outcomes of engineering decisions, in our case related to security outcomes. Effective predictive models closely connect *synthesis and design*, on the one hand, with *observation, analysis, and evaluation*, on the other. Indeed, in the development of mathematically-based reasoning systems, including language and architecture frameworks as well as associated semantic models, these two sets of activities are inseparable.

The broad landscape of the science of security precludes comprehensiveness by the Lablets. Therefore, the portfolios of projects undertaken focus on a set of specific hard technical problems in cyber security that are significant to practice and that are especially likely to benefit from and strengthen the foundations of a scientific approach to the science of security. The structure and organization of the Lablets under the guidance of the NSA drives collaboration on the advancement of explicitly scientific approaches to these hard problems. This focus and collaboration promotes the rapid creation of stronger evidence of the technical validity and potential operational validity of results, and thus hastens their transition into practice. By coordinating across the multiple efforts both within the Lablets and undertaken by collaborators in the broader community, we will develop fundamental principles and methods that guide the design and implementation of secure systems despite the ever-changing security landscape.

We summarize in this paper a selected set of five of the "hard problems" in the science of security. We selected these problems because of their level of technical challenge, their potential operational significance, and their likelihood of benefiting from emphasis on scientific research methods and improved measurement capabilities. We aimed for the hard problems to be crisply stated and well scoped, to assist us in assessing progress towards solutions. Solutions may have the feature of *incrementality*, in that discernible steps will lead towards an overall solution, each step with the potential to result in a corresponding increment of mission impact, even when a fully comprehensive solution may remain elusive. We note, finally, that some Lablet projects aim to make progress in areas of security science not covered by these hard problems.

The focus on advancing scientific practice is driving us to be explicit about the scope of problems, the precise mathematical formulation of those problems, and the processes of hypothesis formation, data gathering, analysis of those data, and, where appropriate, development of rigorous proofs derivable from the models and problem statements. That explicitness can have significant benefits in supporting appraisals of the operational significance of work, and thus in transitioning research results into practice.

But that explicitness also presents challenges. The modeling of security-related phenomena is often characterized by a framing difficulty relating to the scoping, or *expressiveness*, of abstractions. A technical research problem is framed from an operational problem through the selection or invention of abstractions. Abstractions enable focus on salient aspects of a problem by omitting certain real-world details not pertinent to the problem. The abstraction process enables the creation of feasible but also precise models and the attainment of technically valid solutions. But these solutions are framed in the context of the abstractions, and poor choices in abstraction can detract from the operational significance of these results. Making this choice-making more explicit can greatly assist in the understanding of whether a technically sound result also has sufficient operational soundness to be adopted in practice—and what additional operational constraints may be required to lend validity to the scoping of the abstractions on which the technical results are based. As a result, even when modeling is based on seemingly well-established abstractions, we should address not only technical validity, but also issues of operational validity related to the scoping of the abstractions. Often the abstractions are selected on the basis of convenient meta-theoretic properties; we must nonetheless continually reassess how the abstractions might be adapted to enhance operational validity.

In the context of the modeling and mathematical work, we propose five dimensions in advancing the science—to creating mathematical models that are more valuable than those already in use.

The first dimension of value is the *breadth of scope* of operational validity—a valuable model will have fewer caveats in the predictions that result from the mathematical reasoning that is associated with the model. Put simply, the model applies to a broader range of security attributes, both in the context of design and in the context of evaluation or observation.

A second dimension of value is *scalability*—the degree of capability and extent of complexity of artifacts that can be feasibly modeled. This second dimension is traded off with the first dimension—a valuable model may be more narrowly focused in its scope, but may more readily support scaling up, composition, and efficient practices for development and evaluation.

A third dimension of value is the ability to *hybridize* with other models, which can greatly enhance operational scope and validity. Research activities in the Lablets push in all three dimensions.

A fourth dimension of value is the "*analytic capacity*" of candidate models to support effective and efficient formal reasoning that can feasibly lead to valid and useful predictions. This capacity for formal reasoning goes well beyond basic computability characteristics such as asymptotic complexity or decidability. Indeed, many of the formal systems built into analysis tools are intractable or undecidable—but in many practical cases the computations nonetheless turn out to be feasible to compute. (A classic example is Hindley-Milner type checking, which is computationally intractable—exponential in the worst case—but near-linear in practice.) Indeed, when development is intertwined with modeling and analysis, development practices can often be adjusted to accommodate the computational limitations of the models.

A fifth dimension of value for models is *usability*. Human developers and evaluators use metaphors or intuitive associations with the models with which they interact. Sometimes the models connect well with metaphors already familiar to developers and evaluators, and sometimes the models may require human users to climb a learning curve to assimilate the associated metaphors; in this case valuable models will support a learning process for the target population of human developers and evaluators. In many cases, however (including the Hindley-Milner example), much of the mathematical depth and complexity can be hidden or encapsulated behind the surface expression of the models, enabling simpler and more intuitive associated metaphors. Success in this regard can enable us to develop models that are mathematically deep but nonetheless feasible for use by a broad population of engineers and evaluators.

Science rests on both theory and experiments. Theoretical security work (such as the development of formal systems for reasoning about mathematically-based models) has historically been more rigorous than experimental work. To strengthen the rigor of experiments (and provide means of validating theory), we need additional efforts in modeling and security metrics/measurement. Both are essential in the evaluation of hypotheses and also in assessing validity of results: evaluating the operational significance of results developed within an abstracted scope. Measurement related to cyber security, however, is decidedly fraught with uncertainty, and much of the effort is devoted to enhancing the evaluative capability, even when purely quantified outcome measures continue to elude development. Progress in modeling, measurement, and experimental design appears to require acceptance of the reality of multiple criteria and dimensions of evaluation, with overall weighting among these dimensions based on information about the operating environment and threat. This weighting enables incremental progress, as noted above and helps motivate adaptation of system design when particular critical criteria cannot be fully addressed.

On the theoretical side as well, much remains to be done. Classical formal methods have to be extended to encompass details that experimental validation requires, and to scale up to the sizes of practical systems. As we apply methods of control theory and stochastic analysis in the security realm, we find that new models and analytic approaches are needed to describe the richness of real security contexts (and enable validation), and that new algorithmic approaches are needed to deal with the combinatorial complexity of typical solutions.

The Lablet process incorporates a diversity of efforts and explicit collaboration to synergize common technical ideas and approaches to support advances in the identified five hard problems. The process will also address the common goal of making our scientific process more explicit for the purpose of enhancing validity and accelerating impact of research results on practice in development, evaluation, and operations.

For each of the hard problems, we measure our progress in several ways, beyond the usual technical validation through scientific community processes. Particular aspects of technical and operational validity are addressed in each of the five sections below. Some common features include (1) development of new models whose technical validity can be ascertained and that have measurable impact on the potential to construct, assure, and operate secure systems; (2) refinements of existing models and their associated validation approaches; and (3) improved means to support validation, including empirical operational validation of models, increasing our confidence that the models are responsive to real needs and threats.

The five hard problems are detailed in the sections below.

1. Scalability and Composability

Challenge:

Develop methods to enable the construction of secure systems with known security properties from components with known security properties, without a requirement to fully re-analyze the constituent components.

Introduction:

As systems grow in complexity and size, the challenge of reasoning about security attributes seems to grow even more rapidly. Indeed, it is generally accepted that when separate components are aggregated into larger systems, the entire assembly must be fully evaluated to reach an overall assurance judgment. The need for full system evaluation may exist even when there are established evaluation results for the individual components, since component evaluation results may not fully account for the many kinds of interactions among components. For that reason, assurance practices—in general—do not readily scale up with system size and complexity; any increase in scale can result in a disproportionate increase in the difficulty of evaluation in support of an assurance judgment.

The challenge is exacerbated by the reality of most modern systems, which make use of larger numbers of components that are provided through more diverse and complex supply chains to create and sustain those components. The diversity of sourcing, nearly unavoidable for systems that involve Web services, mobile devices, big-data analytics, and other established socio-technical ecosystems, suggests that assurance cases for complex multi-component systems must increasingly rely on *direct evaluation of the product*. That approach is a departure from current practices, which rely more heavily on *trust in the provider* and on *process compliance.*

Science that is known in the area:

For certain particular attributes, we know that it is possible to develop component interface designs and evaluation methods that afford a property of composability. *Composability* means that conclusions about individual components can be combined into an overall conclusion for an aggregate system, for that particular attribute, without the necessity of revisiting the analyses of the individual components.

Generally speaking, composability is the principal gateway to scale, enabling us to build large-scale extensible systems with sound assurance arguments at practical cost. That is, composition is a key to productivity for both the developers and the evaluators of systems. Success in composition derives from the particular ways that security and quality attributes are modeled and analyzed.

Some of the most significant breakthroughs for composition relate to type systems. The kind of object-oriented typing used in Java, Ada, and C#, for example, features a dynamic and composable approach to assuring type conformance and data integrity in separately developed and compiled components (modulo compiler and runtime correctness). It is important to note that in the years prior to the emergence of these languages, there were questions regarding whether this kind of composability was technically feasible. Type systems are continuing to become more expressive, making it possible to make broader assurance claims on the basis of the more advanced analysis techniques associated with the advanced type systems.

Composability has also been developed for a large number of critical modeling attributes for which there are static analysis techniques, including properties ranging from aliasing and effects to various concurrency properties, typestates, access permissions, and others. Those results have significant potential benefit in assuring critical attributes in the increasingly sophisticated frameworks used in the growing population of Web applications and "framework and apps" ecosystems. There have also been composition results related to other semantic properties, as well as protocol and API design, including cryptographic protocols.

Recent Lablet results include system-level models for reasoning about compositions of components that have probabilistic or potentially adversarial behaviors. Compositional reasoning techniques are emerging for systems with larger and potentially opaque components; they are based on trace properties and algebraic expressions of security properties, This recent work, in the CMU Lablet (Datta, Wing, Jia, and Harper), supports modeling of Web-based systems in the presence of adversaries and includes the use of trace properties and language-based techniques to support a "blame semantics" that can link composition failures with flaws in the modeling of particular system components. In order to provide a more scalable approach, these approaches must combine state-transition modeling (trace properties modeled through abstractions such as Clarkson and Schneider's hyperproperties) with structural modeling (focused on component structure, interfaces, and flows across the component boundaries).

Research that needs to be done to meet the challenge:

Composition issues exist at nearly every level of software structure, ranging from large-scale frameworks and subsystems, to APIs and architecture, to individual code elements. Additionally, composition issues and challenges vary according to the particular quality attribute and supporting analytic property being assessed. There are likely hundreds of different such properties. This implies that any systematic pattern or approach to composability could have extraordinary impact, comparable to that of first-class typing in programming languages.

Composability is, in general, difficult to achieve for many security-related attributes, such as those related to confidentiality and locality of data (e.g., confinement of data references to a particular set of components or threads), integrity of data (affording read access while protecting write access), and availability of services (including internally within a system as well as in service-oriented models). But there is a growing set of approaches to modeling, analysis, and language, and they are offering promise. Examples include concurrency properties in typed languages (e.g., [1], [2]); underlying static analyses related to aliasing, effects, and the like; and critical API-related properties such as permissions and type states (e.g., [3]), among others. That set is growing as a consequence of often deeply technical breakthroughs in the underlying theory and science, which inform particular design decisions in the development of modeling and analytic tools. Those tools, in turn,

inform the design of programming and evaluation practices as well as the languages and tools that support them.

The specific technical challenges in this area closely follow the dimensions of improvement outlined in the introduction for mathematically-based models. The models, as suggested earlier, must focus both on properties associated with the set of possible state transitions in a systems and also on structural properties of a system, most specifically its component structure. In both of these dimensions, there are often clear expectations that a designer or security evaluator might express. This might translate into liveness or safety properties associated with patterns of state transitions. It might also translate into information-flow properties among trusted and less-trusted components in a complex system.

As noted, one of the benefits of the Lablet model is that we can identify common features and characteristics of models that do or do not support composition. We aspire not just to accelerate progress towards composable models, but also to understand what kinds of language features, tools, and practices can assist in achieving composition more readily for a diversity of security-related properties that contribute to overall assurance judgments. Achievement of that goal will be a key enabler for the development of larger and more complex secure systems.

That concept of composability encompasses dynamic and adaptive designs; it includes not only static models and structures defined during a development process, but also the capability to scale and compose results from complementary dynamic models and from architectures and tools that support self-adaptation of operational systems.

Progress comes not only in the form of advancing these dimensions of modeling, but also (as noted above) combining or hybridizing the models to permit property expression that is better targeted to developers (i.e., in engineering terms) and evaluators (i.e., in security outcomes) and also that provides usability features such as pithy specifications, incrementality, and the like.

2. Policy-Governed Secure Collaboration

Challenge:

Develop methods to express and enforce normative requirements and policies for handling data with differing usage needs and among users in different authority domains.

Introduction:

Collaboration is complex because the collaborating parties are autonomous and heterogeneous: they face different requirements, exercise different policies, are subject to different laws, and adopt different attitudes, such as to risk. Collaboration exacerbates security risks because of such diversity, especially because the parties involved might inadvertently harm each other even when they are cooperative. A system that enables and enforces secure collaboration among two or more parties must consider (1) models and languages to specify normative standards of correctness as well as policies for different application contexts (e.g., workflows, databases, and, more broadly, delegation of responsibility or authority with respect to shared resources); (2) analysis to determine whether policies (when combined across collaborators and contexts) comply with the stated *normative relationships* (abbreviated as "*norms*" in the remainder of this document) of a collaboration context, that is, are safe and live (e.g., whether privileges would flow out only to certain parties who should have those privileges); (3) enactment (efficient implementation, e.g., via data structures that speed

up authorization checking to determine whether a request should be granted based on the current policy); (4) enforcement (ensuring that collaborating parties do not harm each other despite their autonomy); and (5) trust dynamics among the various parties (especially as it affects their policies for collaboration and is affected by their perceived outcomes of such collaboration).

Science that is known in the area:

Policy-governed collaboration in cyber systems is often handled via access control mechanisms, which selectively determine who can access services and resources and what access should be provided. There are two major models in traditional access control: discretionary access control (DAC) and mandatory access control (MAC). Role-based access control is a relatively recent introduction that enables scalable privilege management in enterprises. Those models are further enriched with additional expressiveness (e.g., support of temporal and context constraints), tailored for specific information systems (e.g., operation systems, workflow systems, delegation in distributed systems, and firewalls), and extended with nontraditional policy components (e.g., usage and obligations). In terms of scientific advances in security policies, aside from the development of formal policy models, a large amount of work has been done on the analysis of security properties of policies and the intrinsic feasibility and complexity of such analysis. Example security properties studied are safety and liveness (i.e., whether an entity may eventually obtain or lose certain privileges). Notable results include the undecidability of checking safety in the Harrison-Ruzzo-Ullman (HRU) [4] model, and various restrictions of the HRU model to make the problem decidable (and further tractable). Similar efforts and results are found in the analysis of other policy models (e.g., in decentralized trust management). Another important line of work is the theory of secure information flow in MAC and, in general, in lattice-based access control, including Denning's formalism of information flow policies, the BLP and Biba models, and general axioms for secure information flow.

Another important line of past work addressed the problem of inference control, especially in the context of statistical databases. The goal was to prevent one from deriving an individual's private information through a sequence of statistical queries. Recent advances in differential privacy were a significant extension to that line of work. However, the problem has not been adequately studied in the context of policy management, that is, whether one can infer sensitive information, given its explicit privileges as defined by access control policies. No theories have been developed regarding the intrinsic difficulty of that problem.

The established approaches model trust as a matter of trusted certificate chains. Though these approaches are popular in practice, they do not address collaboration at all. Other popular approaches today are probabilistic or even heuristic in nature. They assume a fixed and implicit collaboration context in which a central party can track the trust ratings of the interacting parties such that probabilistic judgments and heuristics can be applied. Such approaches do not take into account the subtle kinds of collaboration contexts that arise in practice.

Another body of work deals with the incentives of the parties involved. This work mostly examines the incentives a party has to be honest in its reports regarding the trustworthiness of another party. The idea is to provide incentives that would encourage honest reporting, based on which more accurate judgments of the trustworthiness of a third party can be made. Those approaches do not directly address the challenge of having parties interact with each other in a trustworthy manner.

The idea of developing a normative basis for trust from a technical standpoint is gaining interest. Bruce Schneier's book *Liars and Outliers* [5] introduces some of the conceptual points that place

security and trust as a matter of potential norm violations by defecting parties. There has been a fair amount of work on institutions in economics and political science, e.g., that of the Nobel laureate Elinor Ostrom [6]. However, those approaches do not provide a formal basis for understanding norms and trust with the precision we need in computer science.

Research that needs to be done to meet the challenge:

Existing approaches often assume that a single organization (e.g., a hospital) owns or controls all the resources in question. Managing the policy-compliant access for that single organization may be complex, but does not address the challenges of secure collaboration. The complexity escalates with sociotechnical systems that feature multiple, autonomous stakeholders whose interests are at best imperfectly aligned, where none has authority over the others. Stakeholders must be able to administer or self-govern such systems in a manner that respects their autonomy while causing no harm to themselves or their collaborators.

Need 1: Understanding normative relationships. As we explained above, trust is supervenient upon norms. Accordingly, it is critical to find a way to model the rich relationships among autonomous parties that underlie any trust relationships that they might develop. We propose to do so with an improved notion of norms. Instead of norms in the general societal sense, we think of norms arising between specified parties in a specified organizational context. For example, we would not merely capture the societal norm that private information should not be published, but capture a precise normative relationship that Alice prohibits Bob from sharing with others the private information she provides to him. By clarifying the directionality of the relationship and restricting its scope to Bob's actions, we make Bob accountable with respect to his actions and simultaneously protect him from having to take responsibility for the ill-advised actions of others. Based on this understanding of norms, we can now express evidence and opinions regarding the trustworthiness of one party with respect to another. For example, Alice would not determine that Bob is or is not trustworthy for all purposes, but that he is trustworthy with respect to a particular norm: for instance, that Bob is trustworthy with regard to protecting private information even though he may not be trustworthy with respect to keeping his commitment to apply patches to his computer's operating system.

Need 2: Representing policies for an open world. We treat policies as duals of norms. Each party applies its local policies, which reflect its individual preferences. When the parties are cooperative, they adopt policies that help them satisfy the norms to which they are subject. However, a party (as an autonomous entity) could adopt policies under which it would violate some of the applicable norms. That case captures a situation in which a party may decide to willingly suffer a sanction (e.g., pay a fine) rather than keep a commitment. In other words, the norms are set in a society or accepted in negotiation by the parties themselves (in essence, creating a mini-society), whereas the policies are entirely adopted by an individual party. That is, norms are interactive; policies are local.

Need 3: Analyzing norms and policies. Modeling and representing norms and policies is the first step. We need to be able to reason about those representations to determine whether a given system with its concomitant norms and its members' policies is "secure" as desired. Such reasoning can help determine not only what actions a particular member ought to take in a particular setting, but also whether the system specification as a whole is consistent, whether a particular party's policies do not contravene the norms (compliance), and whether each party has sufficient visibility into the system (vividness).

- *Consistency.* Are the applicable norms mutually consistent? Are the policies of a party mutually consistent?

- *Compliance.* Do a party's policies comply with the applicable norms? Notice that instead of asking whether the policies of all parties put together are mutually consistent, we seek to establish their consistency with the norms. That is, the norms provide an architectural abstraction separating the interactive or social elements from the internal or private elements of each party.
- *Vividness.* Can each party determine whether its counterparties are complying with the norms as they apply to them? The idea is that a vivid system would make it possible for each party to monitor the others and determine whether it needed to complain; in essence, the parties would be able to jointly police the system.

Need 4: Understanding trust from a normative perspective. We need to fundamentally understand the dynamics of trust between collaborating parties. In general terms, we consider how a trustor develops and maintains trust in a trustee with respect to some task for which the trustor relies upon the good intent and competence of the trustee. Trust in real-life settings is rarely binary but arises in different degrees. A trustor would use the level of its trust in a trustee to determine whether and how to collaborate with the trustee. As is well-known, the degrees of trust can be considered in probabilistic terms. However, the rational underpinnings of trust are inadequately understood today. Inspired by dynamical systems theory, metrics can be created for evaluation of various interaction mechanisms with respect to various properties. Does a mechanism enable a trustor to learn the true trustworthiness of a trustee, or are outcomes grossly dependent on initial estimates? How quickly does a mechanism enable a trustor to find the true trustworthiness of a trustee? These properties can be formalized using dynamical systems theory; specifically, they can be formalized in terms of the existence, uniqueness, and stability of fixed points of a best-response function derived from an interaction mechanism. Real-life trustors and trustees, however, are imperfectly rational. We need to study the dynamics of trust via simulations involving models of limited rationality such as quantal response (economics) and prospect theory (psychology).

Need 5: Understanding information flow and inference. Given a set of database access control rules, a fundamental security problem is the need to determine whether sensitive information may be leaked to (or inferred by) unauthorized users. The problem is particularly challenging for fine-grained (row-level or even cell-level) access control, mainly for two reasons: (1) fine-grained access control is much more flexible and expressive than traditional table-based or column-based access control, which permits rather richer and unexpected ways to infer seemingly inaccessible information; and (2) we cannot assume high trustworthiness of entities, especially those from other authority domains. Thus we have to take collusion into consideration, such that entities may combine information accessible to each of them and together infer sensitive information that is inaccessible to any of them individually. How can we determine whether sensitive information can be *deterministically* (rather than probabilistically) inferred?

To deal with that challenge, research is needed to develop a rigorous formalism for specifying fine-grained security requirements and the reasoning capabilities of colluding parties, and to study under what settings information leakage can or cannot be effectively prevented. As for past scientific advances on security policies, the goal is to investigate the intrinsic difficulty of the problem: (1) whether there exists an algorithm such that information leakage can always be detected; (2) if so, whether the problem is tractable; and (3) if not, under what special (but still general enough) policy and system settings the problem becomes decidable (or, even better, tractable).

Need 6. Understanding privacy in big data. Because of the sheer magnitude of big data, privacy control under such circumstances requires us to rethink almost all aspects of privacy protection, including how privacy is modeled, what mechanisms support it, and what evaluation metrics and methodologies are appropriate. Specifically, previous work tends to emphasize technical themes, such as confidentiality guarantees or access control. However, privacy is a consideration only when there are two or more autonomous parties involved. Thus, it would be appropriate to develop a normative account of privacy, wherein a principal would provide (private) information to its collaborators and yet expect the collaborators not to use, store, or divulge that information in ways not essential for the collaboration at hand. A normative model would require decentralized mechanisms for supporting privacy norms as well as methodologies for engineering and evaluating systems with respect to the privacy guarantees they accord different kinds of information of different roles and the acceptability of such guarantees to end users.

3. Security-Metrics-Driven Evaluation, Design, Development, and Deployment

Challenge:

Develop security metrics and models capable of predicting whether or confirming that a given cyber system preserves a given set of security properties (deterministically or probabilistically), in a given context.

Introduction:

Lord Kelvin said, "If you cannot measure it, you cannot improve it." Likewise, until we can measure security, it is difficult to know if we have improved it [7]. However, predictive security metrics and measurements have been shown to be a "tough problem, not to be underestimated" [8]. The scientific background for achieving this capability includes the development of a wide variety of system and threat metrics and modeling techniques. In particular, security metrics and models can inform the:

- selection of software/hardware development processes and practices,
- prioritization of effort towards the areas of the system that indicate the highest security risk,
- decisions about system architecture and design, and
- readiness of a system to be released.

Broadly speaking, measures of security are intended to indicate the degree to which a system can be expected to perform its intended function under particular conditions of operation, including an attack [9]. The metrics must be quantifiable, feasible, repeatable, and objective. Scientifically sound metrics and models must be validated [10].

Security metrics are particularly challenging to develop for many reasons, including the uncertain and variable nature of the:

- behavior of intelligent adversaries,
- intensity of adversarial effort,
- attractiveness of the target system,
- impact of the composition of system development process choices, and
- impact of the composition of architecture and design decisions.

Different systems face different adversaries and attacks, making generally applicable security metrics computation techniques difficult to develop. For example, a growing concern (for which the alarm was sounded at least a decade ago, e.g., [11]) is the possibility of successful cyber-attacks on critical civilian infrastructures and services, such as electrical transmission grids, trains, water supplies, food processing facilities, hospitals, financial institutions, and so on. Government-owned systems and critical infrastructure also attract cyber-attacks. To produce meaningful results, security analyses (automated or human-aided) should include the context of the specific adversaries, since context may offers clues (and solutions) that otherwise may be missed (e.g., [12]).

Science that is known in the area:

Metrics-based analysis has been used to guide the development and deployment of high-assurance systems, but at high cost, and with a focus that has more typically focused on safety and accidental faults, rather than malicious attacks. These challenges suggests that metrics-based analysis techniques need to be rethought before being used for security assessment in order to account for the presence of an intelligent, determined, and opportunistic adversary that may be pursuing an agenda of his or her own.

Science-based work in this direction has been done, but has typically been narrowly focused on particular systems aspects. For example, security metrics have been used in statistical models to predict which files are attack- and/or vulnerability-prone, to help prioritize vulnerability removal efforts. Researchers have also investigated a number of security metrics for use in prediction models, such as reliability-based failure data [13], static analysis alerts [14], developer metrics [15], and software complexity [16]. Additionally, researchers have studied the characterization a system's attack surface; that is, is the set of ways in which an adversary can enter the system and potentially cause damage. Attack surface metrics [17] quantify the degree to which a system can be entered and attacked. Unfortunately, models have shown a low level of statistically significant correlation between the metrics and vulnerabilities, indicating the need for the identification of more metrics to explain more behavior. Security metrics and models must predict discovered vulnerabilities attacks which generally rare events ([18] [19], [20-22]), causing statistical challenges. Currently, in open-source software, only about 0.05% to 5% of the total number of software problems reported/discovered appear to be related to security, and only about 0.05% to 2% of those appear to result in field failures (e.g., [20]).

An example of a more general, but still limited, technique that has been around for quite some time is fault-tree analysis (e.g., [23], [24, 25]). Fault-tree analysis identifies dangerous failure points or faults, and then (working backwards) constructs possible failure paths that can result in such failures. Fault-tree analysis has been applied to security attacks in the form of attack trees. Attack trees (e.g., [26]) have been used to describe how sets of events can constitute a security compromise. However, attack trees do not contain a notion of time, which prohibits expression of attacks as time-ordered sequences of events.

Attack graphs and privilege graphs extend attack trees by introducing state to the analysis. Attack graphs and privilege graphs enable state-based analysis (e.g., [12]). While the list of attack goals and associated trees can never be exhaustive, the question is, are the assumptions correct, and do the trees provide necessary and sufficient coverage to meet the security preservation requirements of a system, particularly a distributed and concurrent system? A suite of adequacy risk metrics needs to be developed and validated if attack trees are to be used in practice.

Adversary-based analysis is the focus of some other system security analysis techniques. It brings in an important active element: the human attacker. For example, Mission Oriented Risk and Design Analysis (MORDA) assesses system risk by calculating attack scores for a set of system attacks. The scores are based on adversary attack preferences and the impact of the attack on the system mission. The Network Risk Assessment Tool (NRAT) examines a set of attributes of the threat actors (adversaries), the attacks, the information system protection (defense), and so on. The ADversary VIew Security Evaluation (ADVISE) method [12], aggregates security-relevant information about a system and its adversaries to produce a quantitative security analysis useful for holistic system security decisions. The ADVISE approach is to create an executable state-based security model of a system. The security model is initialized with information characterizing the system and the adversaries attacking the system. The model then simulates the attack behavior of the adversaries to produce a quantitative assessment of system security strength.

Research that needs to be done to meet the challenge

Many low-level security metrics exist, but the challenge is to produce a quantitative assessment of the security of a system as a whole. Conceptually, this type of assessment requires an understanding of how the components of a system interact within the system, and how successful attacks affect system operation. More specifically, a whole-system security model should contain system information relevant to security mechanisms, including the penetrability of security-enforcing devices (such as firewalls) and the existence of possible connection paths between disparate parts of a system. It should also describe the adversaries threatening the system and the how the behavior of a system's users may affect its security. Likewise, system responses to successful attacks should be considered when predicting overall system security. In short, the system itself, its security mechanisms (both those that avoid successful attacks and those that respond to successful attacks), its envisioned adversaries, and how its user's behaviors can affect overall security must be modeled.

In addition to the models themselves, methods must be developed to quantify, as accurately as possible, input parameter values to the models. Since these models will undoubtedly contain parameter values that will be difficult to quantify accurately, there must be means to perform sensitivity analyses to uncertain input values.

While not exhaustive, the following paragraphs detail examples of research could be done to help achieve these capabilities. In particular, decades of software engineering research has produced empirical evidence of the effectiveness of (*accidental*) fault and failure prevention and removal practices as levers for achieving overall quality objectives. Essentially none of this data exists for vulnerability prevention and removal techniques as levers for achieving security objectives. The Building Security in Maturity Model (BSIMM) [27] provides high-level information on the software security practices of 51 leading software security initiatives. However, metrics and models do not exist to associate the choice of the practices of these initiatives with improved security (i.e., avoidance of or resilience to *intentional* attacks). Kaminsky [28, 29] has shown that improved security practices seem to yield improved security but does not provide metrics on effectiveness of specific practices and/or architectural decisions.

Likewise, Software Reliability Engineering (SRE) [30] is a metrics-based software development practice based on the operational profile, metrics (such as defect removal yield) of the effectiveness of fault prevention and removal techniques, and the use of operational profile and of statistically-valid reliability prediction models. Teams use SRE to choose development practices, the use of which have historically demonstrated the ability to enable teams to achieve their desired failure

intensity objective. A similar practice, Software Security Engineering (SSE) could be developed to be utilized by software development teams to enable a team to, based upon empirical evidence, choose a set of vulnerability prevention and removal techniques that have been shown to enable a team to achieve their desired vulnerability intensity objective and resultant security. Prediction models could then be used to make release-readiness decisions, such as whether a system is predicted to be "secure enough" for deployment.

Finally, the low level of statistical correlation between existing security metrics and vulnerabilities [16] indicates that additional metrics are needed to explain the relationship between software and hardware development artifacts and vulnerabilities. The problem of capturing the dependency of a system on its environment appears at this time to be domain-dependent, as are the sensitivities of a given system's behavior to its environment. The first step is to get experience in capturing these dependencies in a few sample cases, e.g., the vulnerability of secure distributed flight control systems to manipulation of the radio channel. Then, research is needed that can approach the problem more abstractly. Both concrete and abstract work is needed to show how correlation of environmental observations and system activity may reveal misbehavior by the system, and to identify unique attacks on the system that are made by manipulating the environment and/or the sensor data derived from the environment.

4. Resilient Architectures

Challenge:

Develop means to design and analyze system architectures that deliver required service in the face of compromised components.

Introduction:

Experience suggests that given enough time, resources, and talent, attackers can penetrate heavily defended systems. Accepting that fact, a widely recognized challenge is that of developing system architectures that can "tolerate" successful attacks on system components. The prevailing strategies to tolerate attacks include (1) layers that provide "defense in depth" and (2) diversity within and among system components, so that a breach of one component does not immediately enable a breach of others. In developing and identifying scientific methods and fundamental principles for tolerating compromised components, we give a particular emphasis to the practical impact on actual systems. Thus, our efforts include the development of analytic methods that show whether critical system service will remain available (e.g., that some performance or availability criteria will be met), even in the face of some specific degree of intrusion or compromise.

This hard problem depends in part on the metrics hard problem, insofar as the assessment of an architecture's resiliency depends in part on metrics used to quantify that resiliency. For the current problem, the emphasis is on the structure of the system.

Science that is known in the area:

The notion of attack and intrusion-tolerant architectures is well known. The common strategies can be classified into (1) those that contain an attacker within a compromised component (i.e., defense in depth), and (2) those that make the attack harder by changing the target (i.e., diversity). To contain a compromise, secure system architectures frequently follow the principles of privilege separation and least privilege laid out by Salter and Schroeder in 1975 [31]. That is, software architectures, ranging from microkernels to network servers [32, 33] to recent prototype Web browsers

[34-36], minimize the impact of a compromise by using privilege separation and then assigning least privilege to the separated components. An alternative line of research seeks to enable a system to operate through intrusions (e.g., the DARPA OASIS program). For example, the Sitar project relies on detection of intrusions, and on detection of reconfigurations of protected servers. A layered approach forces an attack to push through three levels of protection. Likewise, a number of other projects (DPASA, Willow, DIT, Hacqit) detect and react to intrusions. Such science as exists here enumerates the number of penetrations or ways of penetration needed to completely compromise the system. Another approach to intrusion tolerance relies on protocols and algorithms that tolerate intrusion by replication, voting, and methods derived from Byzantine fault tolerance. Here, the science is in the development and analysis of models that establish limits on what subsets of compromised components a system may contain and still produce a correct result.

A basic element of most of those approaches is that different system components pose independent challenges to the attacker. That strategy is often referred to as *diversity*, and serves a function analogous to that of biological diversity across and within species, which can enable some creatures to survive even when others succumb to disease. To achieve diversity, we can change components in several ways. Software diversity can be leveraged at a macroscale (e.g., with different OSes and network layouts) [37-40] or microscale (e.g., address space layout randomization, instruction set randomization, host-specific compilation, service re-configuration, or runtime adaptation) [41-44].

Research that needs to be done to meet the challenge:

Analytic methods for the design and analysis of system architectures that tolerate intrusions with quantifiable service levels might be developed using the more mature methods of reliability and availability analysis. That said, there will be important differences between new approaches and existing reliability and availability analysis approaches, particularly with respect to the way breaches are modeled and the way we maintain delivery of required service levels in the face of penetrations.

Much work in reliability and availability modeling has been built around the assumption of statistically known failures of hardware and software components. That assumption, along with system structure that uses redundancy, yields analytically tractable models, at least for moderately sized systems. A pressing challenge is that intrusions affect system components in ways that are quite different from random hardware or software failures. An analysis of the impact on system behavior given a set of compromised components can be helpful, but in a naïve approach, it would require consideration of an exponentially large number of sets of compromised components. Research is needed to create analytic techniques that defeat the inherent combinatorial complexities. One line of research could target large, detailed system models, developing statistical methods for estimating the quality of service in the face of potential intrusions. Another line of research could focus on more abstract models, trading detail for tractability in an effort to estimate performance while under attack. Abstract models are better suited for use at design time, to help guide the architecture in directions most likely to yield resiliency and guaranteed delivery of service. Another potentially valuable research focus in this area would be to develop techniques for modeling existing legacy software or software/hardware systems that constitute commonly used practical methods to identify regions of efficacy, and expose any "Achilles' Heel" vulnerabilities.

Cyber attacks and defenses against them have quantifiable objectives. Both attacker and defender decisions have costs and rewards. The interplay between attack and defense is a fertile ground for the application of game theory. Game-theoretic research is needed where system resiliency is a

paramount objective. Substantial developments are needed to mesh the abstraction and idealized assumptions of game theory with the joint challenges of uncertainty and overwhelming detail available from operational systems.

In the meantime, while various defense principles (e.g., defense in depth and diversity) and mechanisms are generally considered good security practices, their presence in practical systems has been limited. For example, privilege separation has gone unused for the sake of performance. Similarly, software diversity comes at the cost of performance and deployability. There is a strong need to incorporate those techniques into actual systems (which often include legacy software) without greater than linear performance overhead. Most importantly, use of known defensive principles will benefit greatly from the availability of rigorous scientific principles that support their use and quantify their utility.

5. Understanding and Accounting for Human Behavior

Challenge:

Develop models of human behavior (of both users and adversaries) that enable the design, modeling, and analysis of systems with specified security properties.

Introduction:

Traditional cyber security design, modeling, and analysis approaches either do not explicitly consider system participants or, if they do, assume participant behaviors will fall into an existing pattern that can be easily classified. Increasingly, accumulated cyber security data indicate that system participants can play an important role in the creation or elimination of cyber security vulnerabilities. Thus, there is a need for cyber security designs, models, and analysis tools that explicitly take into account the actions and decisions of human participants and how they affect the overall security of the system. While cyber security models traditionally focus on the behaviors of attackers and therefore on "anomalous system behaviors," they usually do not explicitly account for the ways in which normal behaviors of the users of a system can create or eliminate system vulnerabilities, possibly without any malicious intent. Thus, the goal of security research in this area is to develop a scientific approach that provides a structured, quantitative, and theoretically grounded means of designing, modeling, and analyzing cyber security problems whose outcomes are influenced by human-system interactions.

Such a capability would help in some real contexts recently encountered (some inspired by a Illinois Lablet-sponsored workshop at Dartmouth College in June 2012):

(1) Spatiotemporal analytics have long been a primary weapon in the fight to secure systems. Techniques such as intrusion detection [45] or denial of service detection [46] are used to secure systems by scanning large corpora of data and identifying patterns that are inconsistent with the intended uses of a system. However, those approaches are not effective against security problems that arise from deceitful or malicious, but entirely normal, uses of systems. For example, a legitimate, authenticated user may intentionally enter erroneous information into a database. Existing security techniques will not be able to characterize such behavior as an attack, because manipulation of database information is an intended use of many systems. Techniques that use semantic analysis [47] may be able to determine whether information is consistent with what it should be; however, semantic analysis can only be used in limited settings where domain information is readily available. Instead, development of the cyber-equivalent of ideomotor actions [48], which are subconscious portrayals of intention, in many cases may prove more useful for differentiating acceptable and syntactically

valid use from unacceptable and syntactically valid use. Further, by using low-level device usage as input into these models, we can characterize cyber-ideomotor actions in a system-independent way to enable broad use in securing systems.

(2) In a real-life example, a hospital that was worried about clinicians' leaving themselves logged in on "Computers on Wheels" installed proximity detectors and timeouts. The result was that employees rebelled and put styrofoam cups on the detectors to defeat them. Could their frustration have been predicted? Could one generate a curve showing "aggregate exposure" versus "aggregate user frustration" for various timeout values?

(3) Systems with security requirements can benefit from an improved understanding of how humans normally behave. For example, continuous authentication systems can determine when an authorized user is logged into a system based on his or her behavioral biometric signature [49]; if that signature changes, the authorized user is likely no longer using the system. As another example, people use email in different ways; if we can distinguish the behavioral profiles of sophisticated computer users from novice users, we can offer the novice users more aggressive anti-phishing measures [50] without bothering more sophisticated users who are less susceptible to phishing in the first place. As a final example, the people who develop secure software normally behave in a rational way (according to behavioral economics [51]), which may lead them to avoid using security tools and practices when they judge the gains to be large but the risks small if they do not use the security tools. If we better understand the factors that go into software developers' calculations about whether to use security tools, we can build tools and structure organizations that change the equation to favor increased use of tools. In sum, an improved understanding of how people normally interact with secure software systems can improve the security of those systems.

(4) Given the documented shortage of information analysts in the United States [52, 53], efforts to optimize the performance of the existing workforce are critical to the continued security of the nation. Because junior analysts have not yet gained the work-related experience of more senior-level colleagues, it is likely that they will miss or misclassify important system information that is needed to make security-related decisions. Thus, it is important to understand how experts in the field behave when faced with such information. What cues are perceived, and, more importantly, how are these factors weighted by analysts tasked with recognizing patterns of user behavior that might indicate a potential security breach? By modeling how more senior analysts visualize and interpret data, it is possible to understand how they process data and engage in security-related decision-making. Rather than wait the years necessary for more junior analysts to learn those techniques, it may be possible to develop training interventions that allow them to utilize the decision strategies of those who have greater domain expertise. The ultimate deliverable could be enhanced and/or more accurate information analysis that results in faster security decisions that can maintain system security.

(5) A common goal of many systems is to "fully utilize the inherent capabilities of an analyst." For example, visualization systems seek to fully exploit a human analyst's visual bandwidth. Other techniques attempt to harness our short-term and long-term memory, or our ability to apply intuition and domain expertise to manage ambiguity and make decisions in the presence of partial information. Modeling those capabilities will also allow us to identify limits on the advantages of new techniques or technology. For example, current smartphone displays are approaching "retina" resolutions, where the size of an individual pixel falls below human visual acuity limits. Increasing the display's resolution without changing its physical size may offer subtle improvements, but it

will not allow significantly more data elements to be visualized, since they would be too small for the visual system to distinguish.

Each of the above examples motivates the development of a science of building, validating, and experimenting with models of humans, business processes, computer systems, and security mechanisms, with the objective of anticipating ways that security mechanisms may cause frustration (and motivate circumvention), may enable novel types of detection of unintended system use, and may quantify the impact on the overall system of humans working to circumvent security.

Science that is known in the area:

Behavioral and cognitive psychology and economics have many theories about human decision-making. (In general, most of them are based on complex economic models or address situations in which limited choices are available in simple, controlled environments.) They include random utility theory, decision field theory, and prospect theory, among others. Much work has also been done on modeling human performance in military and industrial domains. (Most of that work was related to operational performance, e.g., is a task done correctly given various environments.) In very few cases, however, has this work been applied to cyber security design, modeling, or analysis.

Modeling of domain expertise and decision-making is one area that has been explored extensively within the behavioral sciences. For instance, Gary Klein and colleagues ([54, 55]) have developed the Recognition-Primed Decision Model (RPDM) to explain how experts perceive and process cues within the context of diverse multiple environments such as military operations, firefighting, and neonatal intensive care. Through a research approach that involves structured interviews of experts tasked with describing critical events and decision points associated with shifts in situation awareness [56], researchers can understand how experts make use of their domain knowledge to make decisions. Because experts are specifically asked to identify potential errors made by someone with less experience, training interventions can be designed to facilitate the improved performance of junior colleagues [57].

Some existing work has begun to model normal human behavior. One area has been in behavioral biometrics [58, 59], measuring how people use their computing devices, such as their mice, keyboards, or touch screens. Research in another area has been modeling how people adopt new ideas, such as through diffusion of innovation [60] and behavioral economics [51]. Other research has modeled how people perceive security mechanisms, such as how they perceive security warnings [61]. Researchers have also modeled people's willingness to share information [62], which affects how likely users are to leak sensitive information inadvertently or how adversaries can purposefully extract sensitive information. Finally, researchers have begun to model basic cognitive abilities to determine, for example, how the memorability of a password informs its psychological acceptability [63].

Research that needs to be done to solve the challenge:

First, and most fundamentally, research is needed to identify and quantify what human decisions and actions are important to system security. In particular, we need to determine what humans do that is important in the context of security, what factors influence human performance in these areas, and whether these influences can be quantified. Research is also needed to identify and recommend a common language or taxonomy to describe properties of systems that explicitly consider humans.

For example, researchers have developed methods to perform sophisticated task analysis for a given problem domain, including security analytics (e.g., [64]). Behavioral psychology has methods to capture human mental models, which can then be used to describe the step-by-step strategies an analyst uses to solve specific tasks. Deviation from those patterns would suggest unexpected activity within a system, and one that may warrant further inspection. Once behavioral models have been developed, they will need to be evaluated in the context of real analysts and security domains, with a particular focus on identifying commonalities across tasks and users. A final step would be to develop methods to integrate the models into real-world security systems. More specifically, given enough basic provenance information and real-time information about user keystrokes or mouse movements, it is currently possible to identify individual users and their moods (e.g., anger, fear, happiness) through involuntary physiology- and psychology-driven changes in their typing and mouse movement behaviors.

More generally, work should be done to answer the following questions. How can modeling the difference between user goals and other organizational goals provide better insights into human performance and a greater understanding of overall system performance? How can models of that type reveal how changes in security policies sometimes result in counterintuitive and unexpected system performance or a persistent failure to meet organizational goals? How can explicit modeling of human decisions provide better decision-making tools than traditional cyber-human system security evaluation approaches do? Finally, how can we establish an understanding of the relationship between a human's use of a system and the human's behavior or goals in the real world?

References

[1] D. F. Sutherland and W. L. Scherlis, "Composable Thread Coloring," in *Principles and Practice of Parallel Programming (PPoPP)* 2010.

[2] T. Halloran, "Analysis-Based Verification: A Programmer-Oriented Approach to the Assurance of Mechanical Program Properties," CMU PhD Thesis 2010.

[3] K. Naden, R. Bocchino, K. Bierhoff, and J. Aldrich, "A Type System for Borrowing Permissions," in *POPL'12*, 2012.

[4] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in Operating Systems," *Communications of the ACM,* vol. 19, no. 8, pp. 461-471, August 1976.

[5] B. Schneier, *Liars and Outliers: Enabling the Trust that Society Needs to Thrive*. Indianapolis: Wiley, 2012.

[6] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge, UK: Cambridge University Press, 1990.

[7] S. Bellovin, "On the Brittleness of Software and the Infeasibility of Security Metrics," *IEEE Security and Privacy,* vol. 4, no. 4, p. 96, 2006.

[8] W. Jansen, "NISTIR 7564, National Institutes of Standards and Technology Report, "Directions in Security Metrics Research"," April 2009, http://csrc.nist.gov/publications/nistir/ir7564/nistir-7564_metrics-research.pdf.

[9] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollman, "Towards Operational Measures of Computer Security,," *Journal of Computer Security,* vol. 2, no. 2/3, pp. 211-230, 1994.

[10] A. Meneely, B. Smith, and L. Williams, "Validating Software Metrics: A Spectrum of Philosophies," *ACM Trans. on Software. Engineering Methodologies,* vol. 21, no. 4, 2012.

[11] K. Sullivan, J. C. Knight, X. Du, and S. Geist, "Information Survivability Control Systems," in *Twenty-first International Conference on Software Engineering*, Los Angeles, CA, May 1999.
[12] E. LeMay, M. D. Ford, K. Keefe, W. H. Sanders, and C. Muehrcke, "Model-based Security Metrics using ADversary VIew Security Evaluation (ADVISE)," in *8th International Conference on Quantitative Evaluation of SysTems (QEST 2011)* Aachen, Germany, Sept. 2011, pp. pp. 191-200. https://http://www.perform.csl.illinois.edu/Papers/USAN_papers/11LEM01.pdf.
[13] M. Gegick, P. Rotella, and L. Williams, "Toward Non-Security Failures as a Predictor of Security Faults and Failures," in *International Symposium on Engineering Secure Software and Systems (ESSoS) 2009*, Leuven, Belgium, 2009.
[14] M. Gegick and L. Williams, "Correlating Automated Static Analysis Alert Density to Reported Vulnerabilities in Sendmaik," in *Metricon 2.0*, Boston, 2008.
[15] A. Meneely and L. Williams, "Strengthening the Empirical Analysis of the Relationship between Linus' Law and Software Security," in *Empirical Software Engineering and Measurement (ESEM) 2010*, Bolzano-Bozen, Italy, 2010, p. Article No. 9
[16] Y. Shin, A. Meneely, L. Williams, and J. Osbourne, "Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities,," *IEEE Transactions in Software Engineering,* no. p. to appear, 2011.
[17] P. Manadhata and J. M. Wing, "An Attack Surface Metric," School of Computer Science, Carnegie Mellon CMU-CS-05-155, July 2005.
[18] T. Zimmermann, N. Nagappan, and L. Williams, "Searching for a Needle in a Haystack: Predicting Security Vulnerabilities for Windows Vista," in *International Conference on Software Testing, Verification, and Validation (ICST) 2010*, Paris, France, 2010, pp. 421-428.
[19] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright, J. Dobson, J. McDermid, and D. Gollman, "Towards Operational Measures of Computer Security," *Journal of Computer Security,* vol. 2, no. 2/3, pp. 211-230, 1994.
[20] P. Anbalagan, "PhD Dissertation: A Study of Software Security Problem Disclosure, Correction and Patching Processes," in *North Carolina State University Department of Computer Science*, 2011.
[21] P. Anbalagan and M. Vouk, "Student paper: on reliability analysis of open source softwarefedora," in *19th IEEE International Symposium on Software Reliability Engineering* Seattle, WA, 2008.
[22] P. Anbalagan and M. Vouk, "Towards a Bayesian Approach in Modeling the Disclosure of Unique Security Faults in Open Source Projects," in *21st International Symposium on Software Reliability Engineering 2010 (ISSRE 2010)* San Jose, 2010, pp. 101 – 110.
[23] C. Ericson, "Fault Tree Analysis - A History," in *17th International Systems Safety Conference.*, 1999, pp. http://www.fault-tree.net/papers/ericson-fta-history.pdf. Retrieved 2010-01-17.
[24] M. R. Lyu, *Handbook of Software Reliability Engineering*: McGraw Hill, 1996.
[25] M. R. Lyu, "Software Fault Tolerance," in *Trends in Software*: Wiley, 1995.
[26] B. Schneier, *Secrets and Lies: Digital Security in a Networked World*: John Wiley & Sons, 2004.
[27] J. West, G. McGraw, and S. Migues, "Building Security In Maturity Model," 2012, http://www.bsimm.com.

[28] D. Kaminsky, "Fuzzmarking: Towards Hard Security Metrics For Software Quality?," March 11, 2011, http://dankaminsky.com/2011/03/11/fuzzmark/.
[29] D. Kaminsky, M. Eddington, and A. Cecchitti, "Showing How Security Has (And Hasn't) Improved, After Ten Years of Trying," March 11, 2011, http://www.slideshare.net/dakami/showing-how-security-has-and-hasnt-improved-after-ten-years-of-trying.
[30] J. D. Musa, *Software Reliability Engineering: More Reliable Software Faster and Cheaper*, Second ed. Bloomington, Indiana: Authorhouse, 2004.
[31] J. Saltzer and M. Schroeder, "The Protection of Information in Computer Systems,," *Proceedings of the IEEE,* vol. 63, no. 9, pp. 1278-1308, 1975.
[32] D. Brumley and D. Song, "Privtrans: Automatically Partitioning Programs for Privilege Separation.," in *USENIX Security Symposium*, 2004.
[33] N. Provos, M. Friedl, and P. Honeyman, "Preventing privilege escalation.," in *2th USENIX Security Symposium*, 2003.
[34] D. Akhawe, P. Saxena, and D. Song, "Privilege Separation in HTML5 Applications," in *USENIX Security Symposium*, 2012.
[35] C. Grier, S. Tang, and S. T. King, "Secure web browsing with the OP web browser," in *IEEE Symposium on Security and Privacy*, 2008.
[36] H. Wang, C. Grier, A. Moshchuk, S. T. King, P. Choudhury, and H. Venter, "he Multi-Principal OS Construction of the Gazelle Web Browser," in *USENIX Security Symposium*, 2009.
[37] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser, "N-variant systems – A secretless framework for security through diversity. ," in *15th USENIX Security Symp*, August 2006.
[38] D. Gao, M. K. Reiter, and D. Song, "Beyond output voting: Detecting compromised replicas using HMM-based behavioral distance," *IEEE Transactions on Dependable and Secure Computing,* vol. 6, no. 2, pp. 96-110, April-June 2009.
[39] J. Reynolds, J. Just, E. Lawson, L. Clough, and R. Maglich, "The design and implementation of an intrusion tolerant system," in *Int'l Conf. Dependable Systems and Networks (DSN)*, 2002.
[40] E. Totel, F. Majorczyk, and L. Me, "COTS diversity based intrusion detection and application to web servers," in *Eighth Int'l Symp. Recent Advances in Intrusion Detection (RAID)*, 2005.
[41] E. G. Barrantes, D. H. Ackley, T. S. Palmer, D. Stefanovic, and D. D. Zovi, "Randomized instruction set emulation to disrupt binary code injection attacks," in *10th ACM Conference on Computer and Communications Security*, 2003.
[42] S. Bhatkar, D. DuVarney, and R. Sekar, "Address obfuscation: An efficient approach to combat a broad range of memory error exploits," in *12th USENIX Sec. Symp*, Aug. 2003.
[43] G. Kc, A. Keromytis, and V. Prevelakis, "Countering code-injection attacks with instruction-set randomization," in *10th ACM Conference on Computer and Communications Security*, 2003.
[44] H. Shacham, M. Page, B. Pfaff, E. Goh, N. Modadugu, and D. Boneh., "On the effectiveness of address-space randomization," in *11th ACM Conference on Computer and Communications Security*, 2004.
[45] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering,* vol. 13, no. 2, pp. 222-232, February 1987.

[46] T. M. Gil and M. Poletto, "MULTOPS: a data-structure for bandwidth attack detection," in *10th conference on USENIX Security Symposium (SSYM)*, 2001, p. 3.
[47] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science,* vol. 41, no. 6, pp. 391-407, 1990.
[48] R. Bandler and J. Grinder, *Frogs into princes: Neuro linguistic programming*: Real People Press, 1979.
[49] A. Jain, L. Hong, and S. Pankanti, " Biometric identification," *Communications of the ACM,* vol. 43, no. 2, pp. 90-98, 2000.
[50] H. Kelley, C. K.W., C. B. Mayhorn, and E. Murphy-Hill, "Something Smells Phishy: Exploring the Definitions, Consequences, and Reactions to Phishing," in *Meeting of the Human Factors and Ergonomics Society*, 2012.
[51] D. Kahneman, "Maps of Bounded Rationality: Psychology for Behavioral Economics," *The American Economic Review,* vol. 93, no. 5, pp. 1449-1475, 2003.
[52] G. W. Dickson, Benbasat, I., & King, W. R. (1982). . DATABASE, 1-12. , "The MIS area: Problems, challenges, and opportunities," *ACM SIGMIS Database,* vol. 14, no. 1, pp. 7-12, 1982.
[53] G. R. Mitchell, *America's New Deficit: The Shortage of Information Technology Workers*. Darby, Pennsylvania: Diane Publishing, 1997.
[54] G. Klein, *Sources of Power: How People Make Decisions*. Cambridge, Massachusetts: The MIT Press, 1998.
[55] C. E. Zsambok and G. Klein, *Naturalistic Decision Making*. Mahwah, New Jersey: Lawrence Erlbaum Associates, 1997.
[56] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors,* vol. 37, no. 1, pp. 32-64, 1995.
[57] G. L. Kaempf, G. Klein, M. L. Thordsen, and S. Wolf, "Decision making in complex naval command-and-control environments," *Human Factors,* vol. 38, no. 2, pp. 220-231, 1996.
[58] T. Barik, B. Harrison, D. L. Roberts, and X. Jiang, "Spatial Game Signatures for Bot Detection in Social Games," in *Eighth Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2012)*, Stanford, California, 2012.
[59] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM,* vol. 43, no. 2, pp. 90-98, 2000.
[60] E. Rogers, *Diffusions of Innovations*. New York: Free Press, 2003.
[61] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "The Emperor's New Security Indicators," in *IEEE Symposium on Security and Privacy*, 2007, pp. 51-65.
[62] A. Acquisti and R. Gross, "Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook," *Lecture Notes in Computer Science,* vol. 4258, no. pp. 36-58, 2006.
[63] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: empirical results," *IEEE Security and Privacy,* vol. 2, no. 5, pp. 25-31, 2004.
[64] M. R. Endsley, B. Bolte, and D. G. Jones, *Designing for Situation Awareness: An Approach to User-Centered Design*: CRC Press, 2011.