Networking and Information Technology Research and Development

NITRD and SCORE Workshop Series

Designing-In Security: Current Practices and Research Needs

July 1-2, 2013

Software Engineering Institute

4401 Wilson Boulevard, Suite 1000, Arlington VA 22203

WORKSHOP REPORT

Contributors:

Celia Merzbacher, Ronald Perez, William Scherlis, Tomas Vagoun, Claire Vishik, Angelos Keromytis, Lisa Coote, et. al.

Table of Contents

Executive Summary

In addressing the need for improved Designed-In Security (DIS) research and practice, the Federal Networking and Information Technology Research and Technology[1] (NITRD) Program and the Special Cyber Operations Research and Engineering Interagency Working Group[2] (SCORE IWG) have begun conducting a series of small, invitational workshops with the aim of placing leading researchers in direct contact with leading practitioners to ensure that future research targets those underlying problems that truly limit the practice. NITRD and SCORE have adopted a multidisciplinary approach whereby a broad range of experts are included in the workshops to address the various problems and solutions that may have previously gone overlooked as a consequence of an overly narrow focus. The workshops offer an opportunity for practitioners to become more familiar with research concepts to address their current needs and, similarly, for academics to gain familiarity with operational challenges as well as better identify educational needs and approaches in building a workforce capable of designing and producing higher assurance systems. The innovative ideas identified in the first workshop will be developed and evaluated in subsequent workshops of the series.

Introduction

Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program[3] prioritized Designed-In Security (DIS) as a research theme to foster research that:

> *Builds the capability to design, develop, and evolve high assurance, software-intensive systems predictably and reliably while effectively managing risk, cost, schedule, quality, and complexity. Promotes tools and environments that enable the simultaneous development of cyber-secure systems and the associated assurance evidence necessary to prove the system's resistance to vulnerabilities, flaws, and attacks. Secure, best practices are built inside the system. Consequently, it becomes possible to evolve software-intensive systems more rapidly in response to changing requirements and environments.*

---

[1] http://www.nitrd.gov

[2] SCORE IWG is the U.S. Government's forum for coordinating cybersecurity research activities related to national security systems.

[3] http://www.nitrd.gov/SUBCOMMITTEE/csia/Fed_Cybersecurity_RD_Strategic_Plan_2011.pdf, National Science and Technology Council, December 2011.

Identifying research objectives requires an understanding of current problem areas, successes, and lessons learned by developers of high assurance software and hardware systems. Of critical importance is understanding what factors limit our current state of practice as well as what we can infer about the future of Designed-In Security.

The "Designed-In Security: Current Practices and Research Needs" workshop was held July 1-2, 2013 at the Software Engineering Institute in Arlington, Virginia. The workshop focused on the IT hardware and software sectors, and posed the following questions to the participants:

- What procedures are in use in your industry now for designing in security?
- What processes do you use to identify and validate the best practices in use or that are contemplated for use in your organization?
- What approaches for Designed-In Security in, beyond those currently in use, would you advocate are ready for industry adoption?
- For each such practice, what is the evidence (in terms of effectiveness, resource cost, scalability, usability, and other criteria) to support its use?
- What hard research problems are in most urgent need of solutions?

The workshop opened with leading practitioners discussing the current state of the practice in DIS, followed by a discussion by leading researchers on the state of the research in DIS. After the opening discussion, the participants were organized into three groups: Business Case, Software, and Hardware. The following sections provide summaries of the discussions and the breakout groups.

State of the Practice

The workshop began with two presentations on the state of the DIS practice by Steve Lipner (Microsoft) and Mary Ellen Zurko (Cisco). Both presenters noted that it has not been until recently that commercial vendors have worked towards developing system architectures that enhance security attributes. Efforts remain hampered by a number of factors and a lack of concrete, transferable metrics that have prevented decision makers from shifting their organization towards a more DIS-centric approach. One such factor is the challenge of integrating security-related activities into modern development processes, particularly agile processes. At larger scales, architecture design is a more explicit activity, with a stronger reliance on the technical capability and professional reputation of security architects. In contrast,

at the small scale, agile processes—as currently implemented—may tend to de-emphasize the importance of up-front consideration of architecture, thereby thwarting secure design.

Another factor that plays a large role in how strongly DIS is emphasized is the perspective of the customer. Customers with a higher regard for security are more likely to place the same level of emphasis on the security requirements as they do on the functional requirements; thereby conveying Designed-In Security as a project deliverable to the vendor. However, many customers who have longstanding relationships with their suppliers work under the assumption that the appropriate security features and characteristics will be inherently included by the vendor in the final product. As a consequence, discussions concerning particular security attributes, measurement of these attributes, and the consequences of incidents are relatively limited. Furthermore, standards and compliance regimes tend to focus on process compliance and checklists that outline particular known attacks. As a result, it is difficult to justify a significant increment of cost or delay when customers cannot readily assess the benefits in a direct way, even when providers and clients have a strong trust relationship.

As researchers and practitioners begin designing future security technologies, heightened visibility for security and advanced measurement capabilities (to improve the capacity to assess security attributes of systems during development) are critical in persuading those who are capable of influencing widespread DIS adoption. In assuring security attributes early on, it is also necessary for both the designer and user to understand the operating environment as well as such factors as the potential hazards, potential vulnerabilities, nature of the threats, and the technical characteristics of the software and hardware components of the system. With these considerations in mind, designers will be able to identify security-related technologies and interventions that can support a business case for adoption.

In the past decade, Microsoft advanced an aggressive agenda of reworking its development practices and achieved widespread adoption of these practices by internal development groups. Recognizing that an early focus on security can be greatly beneficial, Microsoft took explicit steps to advance security-related interventions as early as possible in the development lifecycle, as is evident from the Security Development Lifecycle[4] (SDL) process framework. Microsoft was not only responding to the challenge of how to design "good code to begin with," but also the economic consequences of handling defects late in the lifecycle.

[4] *The Security Development Lifecycle*, M. Howard & S. Lipner, Microsoft Press, May 2006.

Among Microsoft's interventions, the SDL has perhaps had the greatest impact on industry practices outside of Microsoft, as is evident from published Building Security In Maturity Model[5] (BSIMM) results.

At Microsoft, choices of programming language are seen as important drivers of outcomes. In particular, deficiencies in C and C++ are largely responsible for a growing industry focused specifically on code-level defects and vulnerabilities. Language improvements, such as strong typing, which advanced into mainstream languages twenty years ago, can make a difference. In order to be more effective in both software design and code synthesis as well as analysis of code and artifacts, improvements must be made to the methods and tools used by the development teams in conducting these tasks. This is necessary for both products and services, in the sense that software is a service.

With respect to the processes and the timing of interventions, advances such as SDL are becoming generally accepted and adopted as a reference model. According to practitioners, models, such as SDL, provide a comprehensive identification of security-related activities during development, as well as experience-based guidance regarding which activities to undertake at which points in a development process. SDL addresses a broad range of threats related to *S*poofing, *T*ampering, *R*epudiation, *I*nformation disclosure, *D*enial of service, and *E*levation of privilege, otherwise known at Microsoft as STRIDE. The BSIMM assessment, developed and administered by Gary McGraw and collaborators, evaluates organizations by comparing their actual practices with those best practices identified by models such as SDL. However, despite the emerging consensus regarding identified best practices, there is nonetheless relatively little science or data to support specific claims regarding security outcomes. This is yet another factor contributing to the "measurement gap" that needs to be addressed with further research.

With respect to tooling, practitioners recognize that the growing variety of process-support tools is, in fact, beneficial and also necessary for nearly all development and sustainment teams. These include Integrated Development Environments (IDEs) for individual developers such as Eclipse, Visual Studio, Rational tools, etc., as well as team tools that support functions such as configuration management of artifacts, automated testing and analysis, issue tracking, and other critical activities. While impact may not be as profound as in process interventions, tooling and analysis also remain encumbered by the "measurement gap." The "measurement

[5] The BSIMM is a study of real-world software security initiatives, http://bsimm.com/

gap" for tooling is exacerbated by the diversity of security related technical quality attributes as well as by the imprecise results of tools used to support analysis and testing.

In education, the challenges concerning DIS range from those encountered by novice end-users confronted with phishing and other social-engineering attacks, to seasoned engineers facing challenges with tooling, analysis, architecture, processes, requirements, and other considerations.

Integrating Practice into Large Organizations

In recognizing the benefits of best practices such as Software Development Life Cycle (SDLC) or SSDL and Microsoft's SDL, the question then becomes how best to integrate these practices into one's organization. Current methodologies are primarily concerned with prescriptive models, unique to each company, that stipulate exactly what needs to be done to ensure the best practices are being fully engaged. As an effective alternative, the creators of BSIMM have created a descriptive model describing what is actually happening in one's organization rather than what should be happening. BSIMM is a descriptive model that can be used to measure any number of prescriptive SSDLs. In essence, BSIMM is a measuring stick capable of applying 122 different measurements to 62 real security initiatives. The underlying idea of BSIMM is to build a maturity model from actual data gathered from 9 well known large-scale software security initiatives. Once the maturity model has been validated by data gathered from 62 different companies, statistical analysis will be performed to determine how effective the model is, which activities correlate with each other, and whether or not high maturity firms appear at similar levels. In time, more activities will be added with changes in security. This tool allows not only a comparison of individual companies to one another but also a comparison of groups/types of companies. Through applying this tool, it was revealed that most companies proceed in the same or similar ways.

State of the Research

The workshop's State of the Research panel was led by John Launchbury (Galois) and included as panelists J.R. Rao (IBM), Michael Reiter (University of North Carolina), Robert Seacord (SEI), and Elaine Weyuker (formerly Rutgers). Panelists currently serve as leading researchers in areas related to Designed-In Security (DIS), including both technical and

empirical dimensions, across the spectrum from basic science to assessment of industrial adoption.

The discussion started with an exploration of barriers to industry adoption of advanced technologies and practices related to DIS. Historically, this has been a significant issue, with persistent difficulties related to Return on Investment (ROI), scaling, analytic accuracy, and performance. Difficulties in calculating ROI relate to difficulties in measurement and also the credibility of claims regarding the value of newly emerging techniques. ROI difficulties also derive from the perceived incremental cost and delay of attending more aggressively to security issues, when the returns on the cost increments are uncertain and difficult to measure. Scaling has been a historical difficulty with both technical and process approaches. Analytic accuracy relates, for example, to the rates of false positives in static analysis tools. Performance relates both to the impact of using advanced tooling on the efficiencies of individual and team practices and also to the overhead associated directly with using the tool, such as from dynamic analyses. Against the background of these historical difficulties is a sense of optimism on all four of these fronts. Much of the discussion at the workshop centered on new research advances, transition successes, and identified transition success patterns that provide evidence in support of this sense of optimism.

Several panelists pointed out the benefits of taking explicit steps to bridge the gap between research and practice, transferring research results to practice when they have overcome barriers of realism, for example, and also transferring from practice to researchers a better understanding of the real problems to be solved and the important acceptance criteria for these problems. Recurring challenges include both size and complexity scaling, barriers in usability by ordinary developers, steepness of learning curve for new technologies and practices, early demonstrations of benefit to technology adopters, and the challenges of making a compelling business case even when other challenges are overcome.

There is a perception of an unavoidable tradeoff between the level of capability and the extent of assured security – that features must be sacrificed in order to ensure a higher level of quality and security. In the discussion, a counterpoint emerged, suggesting that as quality and security practices improve, over the long haul, functionality can actually be increased due to the greater expressiveness of safer high-level abstractions, including understandable security policy concepts. This means moving away from a “penetrate and patch” model and towards security by design, and taking this attitude in multiple dimensions of practice, ranging from audit and

logging to project planning and incident response procedures. It also includes looking closely at the multiple components that comprise large systems and identifying mission critical assets, access paths to those assets, and constructing "fine grained perimeters," such as those that might come through secure processors and root-of-trust architectures.

One example of a transition success and an associated success pattern is the work related to secure coding practices for mainstream languages, including C, C++, and Java. By adopting certain identified concrete coding structures in common situations where naively constructed structures might be dangerous, practicing developers can realize security benefits without much effort. This work has also influenced the ISO C standards process, which has adopted certain small language changes that help developers to more naturally write secure code.

Breakout Sessions

In transitioning to more focused discussions, participants separated into their respective groups of Business Case, Software, and Hardware. Each group was challenged with questions concerning how current practices have come into being; the effects of current practices on stakeholders; limitations of current practices and the possible improvements to those practices; the necessary incentives in achieving those improvements; and the effects of changes on stakeholders.

Breakout Group: Business Case

The Business Case group focused on elements of the business case for DIS as well as research and policies to increase adoption in various sectors. A widely recognized benefit of DIS is that it is less expensive to build in security early on in the design process rather than later in the process or even after the product is deployed in the field. However, in general, difficulties in computing Return on Investment (ROI) has prevented DIS from being implemented on a large scale. Rather than trying to estimate ROI, one participant recommended that companies think of security investment as "insurance" against an increasingly likely and damaging breach. Of course, insurance only pays off if there is a perceived threat. In the Y2K example, companies treated those modifications effected as "insurance," however, after January 2000 those efforts came to a halt.

Participants also identified specific factors that influence a business's decision to pursue DIS. By far the most influential factor is customer demand, which has historically been relatively weak. For the most part, building in security is perceived by the customer as a "cost of doing

business" rather than as an "attribute" or "feature" that contributes to reliability and quality. This is exacerbated by the measurement difficulties discussed above. As security almost never leads to a standalone product security enhancements have not gained the recognition that other, more lucrative features enjoy. However, casting security as an attribute may make its development cost more palatable. Recognition and growing concern of IP theft/loss and the discovery of various side-channel vulnerabilities are motivating the development of security strategies. While changes in government regulations and requirements are also driving investment in security, it was pointed out that government business alone is unlikely to drive industry since it encompasses only a small fraction of most companies' business. Transforming corporate culture into one that supports "security" requires efforts both from top down and bottom up.

A potential trend that could improve security decision making is the movement towards vertical integration in some sectors, whereby more layers of the hardware-software stack are controlled by one company (e.g. Apple's control of hardware and software design). Greater vertical integration allows security to be more targeted and more likely to be cost effective.

The lack of metrics and standards for security also hampers the business case for investment. Standards can be driven by new technology, but typically take time to be developed and adopted. While some claim that security is too subjective to allow for deterministic metrics; the notion of measuring "processes" related to security is generally perceived as both feasible and necessary.

In identifying what research is needed to help form a better business case for DIS in both hardware and software, research concerning techniques that can reduce the cost/time of Designing-In Security was identified as particularly necessary with the caveat that such research should be informed by best practices for incorporating security in different sectors. Advanced research could also be directed towards developing a BSIMM-type model that is more quantitative and scalable, rather than using strictly qualitative processes such as interviews.

In addition to technical research, economic/business research is needed to develop "predictive models" that anticipate the economic impact of security outcomes. Economic models for security should be developed to enable research in creative business models, similarly to what happened with the research concerning the Economics of the Internet in the early 90's. These models will enable developers and businesses to predict the economic outcomes of introducing security features, including requirements for and impact on civil infrastructure. Such research includes developing a large scale simulation of global systems and "systems of

systems." However, in order to support such multidisciplinary research, a common language of agreed-upon security-related definitions and well-defined taxonomies/ontologies are needed.

Other potential research topics that will support the business case for DIS include research into risk/resilience analysis; security and emerging "usage environments" such as social networks; Bring Your Own Device (BYOD); trust relationships among suppliers and customers; and overarching principles that embody economic and security objectives, similar to those developed in the area of "Green Chemistry."

With regards to technology transfer issues, current university research is viewed as insufficiently generalizable and therefore impractical in real systems, although novel ideas can be used internally in the organization's R&D programs. In some cases, a startup can be successful in moving research from the university through the early development phase and, in these instances, venture capital typically plays an important role. However, security technologies are usually supplemental to core features in products, with the consequence that venture capital and other support for incubation and start-up phases are typically weaker than in other technology areas. Despite these weaknesses, often the greatest value of university research as perceived by the private sector is access to faculty and student expertise.

In addition to recommendations of future research areas, participants discussed strategies and policies that could further strengthen the DIS business case. Leveraging existing government-funded programs such as the Small Business Innovation Research (SBIR) and In-Q-Tel were identified as possible mechanisms for facilitating technology transfer. The NSF Science of Innovation and Science Policy program could also support the economic research that is needed to support innovation and the business case for security. The Enduring Security Framework brings together high level industry and government decision makers and could also be leveraged to create high level agreement to support DIS. Building a "community" around secure hardware and software design could be strengthened by an annual workshop similar to the "Economics of Information Security" workshop series. The introduction of such a forum could stimulate increased industry-government-academia collaboration to address both pre-competitive and non-competitive challenges.

Breakout Group: Software

The two primary sets of questions that became the focus of the software working group included: identifying and evaluating current best practices, with a heightened emphasis on (1)

what's working well, (2) where research is needed, and (3) what are patterns of success in adopting development practices and tools that can materially improve security outcomes, and secondly, identifying opportunities and challenges, with a purpose of identifying (1) areas of significance to practice where technical progress could be accelerated, (2) hard problems that offer game-change possibilities but that will require some sustained attention, and (3) mechanisms to accelerate technology transition.

Practice: Requirements. One of the first issues that came up in the discussion was related to security requirements, and specifically how potential security-related requirements can be specified and assessed, beyond expressions such as "keep the bad guys out" or "be able to handle what comes our way." The challenge is how to identify and express particular requirements and then be able to assess their appropriateness to the purpose and operating environment of the system being developed or evolved. The question, in essence, is how we can achieve early validation for particular requirement specifications and models. This is a notable challenge since the threat, operational, and infrastructural environments are constantly evolving and, furthermore, the challenge is exacerbated because many (indeed, most) of the critical technical attributes are not readily testable. Economic drivers are very significant during requirements setting—which for many/most commercial systems is a continuous process over the lifetime of a system. Due to challenges in creating valid economic models, it is often very difficult to navigate the engineering trade-offs other than on the basis of informal experience and expert judgment.

Practice: SDL and Similar Models. The Security Development Lifecycle (cited above) has become a significant positive influence on the culture of development organizations. It incorporates process milestones that include support of modeling and direct evaluation of development artifacts. While the principal focus is on internal measures of process compliance, there are also emerging informal external measures. This is a very significant area where more research is needed. Also, as noted above, the BSIMM evaluation framework defines various key process areas for normative best practice. This enables organizations to assess the extent of their adoption of identified best practices in comparison with identified norms. Although the model is broadly considered to be highly valuable, the connections with results are still informally constructed due to the ongoing challenges of inadequate risk methodology and, more generally, the "measurement gap" for software security attributes and underlying quality attributes.

As discussed in other groups and in plenary, the group emphasized the difficulty of making an effective quantitative business case for security-related interventions in the process of development and evaluation. Due to the "measurement gap," interventions are accepted by senior management in many cases on the basis of technical experience and expertise rather than on the basis of predictive quantitative business models. On the issue of training and education, critical to success is an understanding of the scope and diversity of security-related skills, ranging from system administration activities such as device and platform configuration to architectural design and evaluation. Another issue is how advanced methods are transitioned into practice across a larger enterprise. Most often, there are central teams that capture experience, develop models of costs, benefits, and risks, and can assist internal development organizations in advancing their practice in a way that harmonizes with their problem domain and team culture.

Practice: Modeling and Analysis. The group spent some time discussing particular technical interventions related to modeling, analysis, language, development/evolution data, and architecture. Most significantly, there is a recognition in industry, perhaps most visibly at Microsoft, of the significant value of advanced technical interventions in all these areas—and undertaking those interventions as early as possible in the process.

Technical models and associated mathematical methods for various kinds of analysis are becoming more broadly adopted for a growing set of quality attributes contributing to security outcomes. It is now recognized that a wide variety of models are needed, as is a corresponding variety of analysis methods with varying degrees of formality and rigor. Models are used in requirements formulation, architecture, design, and coding; they are used to predict outcomes with respect to functional features, quality attributes including security, performance, and many other characteristics. Success in any engineering discipline relies on a range of models that are expressive, that cover critical functional and quality attributes, and that can be effectively validated.

The rigor and scalability of analyses associated with particular models can increase naturally over time, with the advancement of the underlying models and, where possible, foundational mathematics. As research proceeds, models become more expressive and useful, while the associated analyses may advance from informal representations, such as diagrams and documents, to formally-based analyses that work in small-scale cases to better analyses that are composable, scalable, and computationally feasible for large systems. This is evident when considering a full range of analytic techniques, model checking, static analysis, type checking,

and verification. The ability to accommodate a range of informal and formal techniques for modeling and analysis is important for success in practice.

As models and tools are developed, there will be a natural advance in the development of appropriate metrics to assess impact despite the fact that there is a considerable lag of metrics behind technical developments and interventions. This suggests that senior managers have come to rely more directly on expert technical judgment in making choices regarding interventions and supporting those choices with some kind of business case. For this reason, the most aggressive adoption is in technology-intensive firms with technology-savvy senior management who understand the issues behind the lag. This highlights the more general issue of how costs, benefits, and risks are assessed for security-related interventions; there is a general perception that some interventions, in the long run, will likely improve productivity in development and evaluation, and sometimes dramatically. However, there are risks and costs associated with any new technology adoption. This highlights the importance of crafting new practices and tools in ways that they can be readily assimilated into practice with minimal incremental cost and risk.

Practice: Programming Language. The choice of programming language is significant. This is, in fact, counter to early folklore that quality outcomes are primarily due to management and process interventions rather than technological interventions. The pervasive modern recognition is that management and process interventions are enabled and enhanced using technology, and that the two approaches are increasingly intertwined. Each language choice, whether it is C, C++, C#, JavaScript, or PHP, has costs, risks, and benefits, and these must be weighed in the context of overall development goals. Additionally, evaluations of these languages must take into consideration not only the technical characteristics of the particular language, but also the associated socio-technical ecosystem, including tools, frameworks and libraries, and human programmer resources.

Associated with languages are identified models and patterns, which include certain "secure coding" practices. Many of these practices have been codified, and these best practices have been explicitly adopted as enterprise standards in technology firms such as Oracle, Cisco, and Siemens. In addition, the practices have influenced the natural evolution of the formal language-definition standards, such as the C-language standards from ISO.

Practice: Architecture. Decisions regarding overall architecture for software-reliant systems are critical. Processes for making and validating architecture choices are both an essential feature of success and a dark (and proprietary) art—the "secret sauce"—in the

commercial world. These choices have tremendous leverage on outcomes related to both quality attributes and sustainability/modernization. A good architecture, for example, can localize and minimize the critical portions of the system that require the largest amount of attention in development and evaluation—enabling a lesser standard to be more safely applied to the larger remaining portions of the system.

Complicating these choices, besides the intrinsic technical difficulties of modeling and analysis, are a range of extrinsic factors including legacy and precedent, domain culture, and organizational and supply-chain structure. Also complicating these choices are architectural design choices, such as those related to resiliency and "moving target" concepts that are directly motivated by the presence of potential adversaries. These include not just adaptation and shape-shifting, but also sensing to detect attacks and configuration damage.

Practice: Data. Contributing to the advancement of all these areas is the dramatic increase in the extent of granular data associated with the software development and evolution processes. This is largely a consequence of the advancement of tool technology, as noted above, and it affords great opportunity to better address the "measurement gap" as well as to afford a possibility of mitigating information loss in development and assuring better control over the configuration integrity of systems and associated artifacts (what Microsoft calls "Managed Code"). Of course, the gap is a moving target, in the sense that technical advancement to close that gap on one end occurs in parallel with improvements in technology and practices that further open the gap on the other end.

Research: Evidence Production. After several decades of slow advancement, there is now more rapid progress and optimism in the research community in a number of areas including languages, tools, models, analysis, architecture, and usability. These advances are enabled by progress in disciplines ranging from foundational mathematics to tool design and human social psychology.

One of the principal research challenges has been the massive information loss that occurs in the development and evolution of complex software-reliant systems. This results in enormous costs in the evaluation of systems, which can involve extensive reverse engineering to rediscover design abstractions and rationale, as well as in sustainment and modernization activities, for similar reasons. The prospects of evidence-based approaches, where evidence in support of security-related claims is amassed during development and sustained in configuration integrity, has been greatly enabled by the advance of modern tooling, modeling, and analysis, all

of which support the creation and capture of massive amounts of development data, as well as supporting the configuration management of this data through the lifecycle as systems and associated models co-evolve.

Research: Usability. It is now recognized that usability by humans—as end-users, as essential parts of an overall system, and as developers and evaluators—must be a major consideration in the advancement of practices. For end users, the design of security-related abstractions must result in metaphors that can be embraced by end users. Certain end user populations can be trained to assimilate the new metaphors, but for others this is not readily possible. For developers, the design of languages, APIs, tools, and models are all influenced by usability considerations. Usability is not the same as simplicity—complexity can be accepted if it can be encapsulated, such as in type systems, or if there is sufficient support by tools and models to help people in managing it.

Research: Hard Problems. The working group recognized some challenges in particular that are in need of sustained attention. Among these challenges include those concerning architecture design, modeling, and analysis. In organizations, there is still a dominant "guru model" where security-critical areas of a system are isolated through canny architectural choice-making, and then handled by experts. A second area of hard problems relates to requirements, and particularly requirements associated with security-related attributes, such as STRIDE, as mentioned earlier. Thirdly, there are the challenges of today's ubiquitously rich supply chains—how to ensure security in systems that are developed by contractors working within arm's-length of the main mission stakeholder.

Research: Technology Transition. The technology transition challenge has several dimensions, one being the advancement of empirical software engineering and science-of-security research to better support the evaluation and validation of hypotheses, including those associated with the claimed benefits of new methods, practices, and tools. A second is measurement, more generally, identified above as the problem of the "measurement gap." Progress in this area would not only facilitate technology transition, but also the management of complex supply chains. A third challenge for R&D managers is tracing the impact of early research investments on outcomes in practices. Several participants noted that the transition pathways are complex and often diffuse, thwarting traceability despite a recognition of the essential role of advances in basic science in achieving major advances the practice. In response to this, the R&D community has undertaken several studies to document both the pathways and

significance of the R&D role. These are more comprehensively described in a series of several studies from the National Research Council.

Breakout Group: Hardware

The hardware breakout group was predominantly represented by industry and academic experts in software and/or systems engineering who had a particular focus in the semiconductor industry. In this regard, the question of how to apply abstraction approaches from the software community to hardware community was one of the first questions posed by participants. As a recommendation for future workshops, participants recommended for the continuation of identifying current limitations of Designed-In Security in the hardware space. Participants argued that DIS has not been more successful because (1) the Return on Investment (ROI) is difficult to calculate and therefore unclear; (2) security is not thought of as a core element of the product; (3) there is a lack of clarity regarding government initiatives; and (4) there is a lack of standard vocabulary/taxonomy. Participants also pushed for addressing these current limitations by exploring high visibility government/industry sponsored challenges and best paper competitions. Topics should be narrowed to a specific problem and addressed by a handful of stakeholder representatives to enable a gap-analysis of specific domains. With regards to format, workshop proceedings should consist of more topic-focused tracks with subject matter expert/session leaders who are all given the opportunity to prepare for the workshop in advance.

One recommendation for easing the introduction of security mechanisms in the hardware was to develop them as dual-use components that have some other useful functionality. Typically, such mechanisms might improve reliability, such as through isolation, or provide debugging or performance monitoring capabilities, such as with the Last Branch Record register or micro-architecture counters.

Regarding recommendations for the state of practice, the group began by endorsing greater diversity from the hardware community, but without sacrificing the software representation since software experience and perspective is always necessary. There is serious potential for improving overall system security through greater collaboration between hardware and software industries. Specifically, there is need for more intense co-design among the hardware and software industries, particularly at the early stages of the software development process. For example, a technique already apparent in the “verticalization trend” is compressing the hardware development cycle so that it better aligns with that of the software development life

cycle. Additionally, techniques and approaches from the software security community are potentially less complex, more bounded and easily applied in the hardware domain. Information flow/taint analysis, composability, abstraction, etc. were all provided as examples.

As a benefit to the hardware industry, research is relatively mature in terms of overall design and transition of research to practice. While security is far from reaching the maturity level of other hardware aspects it would be beneficial to leverage the maturity of hardware to formally define hardware-related security properties and specifications that could drive security verification and other tooling. Hardware aspects to be leveraged include architecture specification, high and low-level design, quality/reliability, verification, widespread use of sophisticated tools, and formal analysis. Participants also observed that current research and best practices are generally isolated to security specific features, capabilities, solutions, and communities such as smart cards and hardware security module ecosystems. There is need for a hardware focused security engineering community, and best practices that are emphasized at all phases of development. Referred to as ***Design for Security,*** participants called for developing hardware equivalents of SDL, BSIMM, etc., and leveraging opportunities to draw from some of the hardware discipline's unique insights such as behavioral analysis, anomaly detection, runtime attestation-like capabilities, authentication, and provenance. The workshop group called for more widespread application of hardware capabilities, to existing challenges and initiatives such as building secure systems from less/unsecure components, moving target defense, tailored trustworthy spaces, reference monitors, etc. The need for further examination and exploration of hardware in vertical markets was also mentioned, with particular emphasis on those markets in which there is a greater need for security such as medical/healthcare, cyber physical systems, and critical infrastructure.

Conclusions

The overall goal for the Designed-In Security Workshop (DIS) is a clear recipe for how we can adapt and enhance the practices in current use for development, sustainment, operation, and evolution of systems, with a goal of supporting an integrated approach to "Designed-In" assurance. This "Designed-In" approach has the potential not only to support more rapid evaluations, but also much higher levels of assurance for complex software-reliant systems. Importantly, this approach lays the foundation for rapid re-certification as systems evolve and are further interconnected with other systems. However, successfully achieving such re-certification

efforts requires a firm grasp of the necessary metrics for evaluating future systems. In reflecting upon which insights would pose the greatest value towards a new approach, sector leaders identified the workshop recommendations they found to be most critical. From the software sector, one of the great benefits of modern tools, and an area where there is opportunity for further advancement, is the retention and exploitation of the large amounts of data associated with modern software production. This data can support the advancement of metrics as well as the production of evidence in support of assurance cases. From the hardware perspective, in addition to more intense co-design with the software sector, leaders highlighted the need for developing security mechanisms as dual-use components that provide additional functional capability. And finally, drawing on a more holistic perspective, business case leaders emphasized the need for a forum that includes industry and government agencies, where a set of diverse, multidisciplinary researchers can share results and ideas, analogous to the annual Workshop on Economics of Information Security.

Contact Information

For more information about this workshop, contact:

National Coordination Office for Networking and Information Technology R&D
4201 Wilson Blvd., Suite II-405, Arlington, VA 22203
Phone: 703-292-4873
Email: nco@nitrd.gov

Appendix A: 2013 Designed-In Security Workshop Agenda

JULY 1, 2013

8:00 am Registration

8:30 – 8:45 Introduction and goals for the workshop

8:45 – 10:15 State of the Practice

Talks by industry representatives laying out the processes in use in their firms to develop a particular product/system and how and where security considerations influenced the design process.

- Steve Lipner, Microsoft
- Mary Ellen Zurko, Cisco

10:15 – 10:30 Break

10:30 – 12:00 State of the Research

Panel by leading researchers on recent results bearing on methods for designing in security, including empirical evidence obtained or needed to support industrial adoption.

- John Launchbury, Galois, Panel Chair
- J.R. Rao, IBM
- Michael Reiter, UNC
- Robert Seacord, SEI
- Elaine Weyuker, Rutgers

12:00 – 1:00 Lunch break

1:00 – 2:30 Breakout Groups: Best Practices

- Breakout Group: Software (Lead: Bill Scherlis, SEI)
- Breakout Group: Hardware (Lead: Ron Perez, AMD)
- Breakout Group: Business Case (Lead: Celia Merzbacher, SRC)

Each group tasked to identify:

- Current best practices for designing in security
- How the practices have come into being
- Effects of the current practices on stakeholders
- What limits current practices, where they might be improved
- What incentives might be required to achieve the improvement
- Effects of changes on stakeholders

2:30 – 2:45 Break

2:45 – 3:30 Plenary Talk

- Gary McGraw, Cigital

3:30 – 4:30 Breakout Groups: Research

Breakout groups continue, shifting to research focus, to identify:

- What research results are available that might advance best practices?
- What evidence is available that these methods would improve practice?
- What research problems are suggested by the forgoing discussions?
- How do you make assessments of the potential of the practice to scale with acceptable risk and cost?

4:30 – 5:15 Breakout Groups: Progress Report

Brief in-progress reports from each of the breakout groups.

JULY 2, 2013

8:30 – 8:45 Introduction to second day

8:45 – 10:15 Breakout Groups: Summary

Breakout groups develop summary characterizations of industry best practices and research agendas.

10:15 – 10:30 Break

10:30 – 12:00 Breakout Groups: Brief-outs

Brief outs by the groups and discussion.

12:00 pm Adjourn

Appendix B: Workshop Participants

Organizing Committee		
Name	**Position**	**Organization**
Martin, Brad	Committee Chair	NSA
Landwehr, Carl	Research Consultant	
Newhouse, Bill	National Initiative for Cybersecurity Education (NICE) Program Lead, Cybersecurity R&D Coordination	NIST
Scherlis, Bill	Professor & Director, Institute for Software Research (SCS/ISR), School of Computer Science	CMU/SEI
Vagoun, Tomas	Cybersecurity R&D Coordinator	NCO/NITRD
Vishik, Claire	Security & Privacy Technology & Policy Manager	Intel
Invitees		
Name	**Position**	**Organization**
Adam, Nabil	Distinguished Professor of Computer Information Systems and Director of Rutgers CIMIC Research Center	Rutgers University
Aitken, Rob	R&D Fellow	ARM
Dill, Stephen	LM Fellow, Center for Cyber Security Innovation	Lockheed Martin
Elder, Matthew	Sr. Manager, Development, Symantec Research Labs	Symantec
Fogerson, Tim	Security Engineering Manager	Intel
Green, Cordell	Director and Chief Scientist	Kestrel Institute
Jaeger, Trent	Professor, Computer Science and Engineering	Pennsylvania State University
Keromytis, Angelos	Associate Professor, Computer Science Department	Columbia University
Kirby, James	SW Engineering Researcher	Navy Research Laboratory
Launchbury, John	Chief Scientist	Galois
Lipner, Steve	Partner Director of Program Management, Trustworthy Computing	Microsoft
McGraw, Gary	CTO	Cigital
Merzbacher, Celia	Vice President, Innovative Partnerships	SRC
Ostrand, Tom	Visiting Scholar, Center for Discrete Mathematics and Theoretical Computer Science & AT&T Labs	Rutgers University

Ozkaya, Ipek	Senior Member of Technical Staff, Architecture Practices	SEI
Perez, Ron	Senior Fellow, Senior Director, Security Architecture Organization	AMD
Rajan, Anand	Manager, Security Research Lab	Intel
Rao, Josyula	Director of Security Research	IBM Research
Reiter, Mike	Professor, Department of Computer Science	University of North Carolina
Seacord, Robert	Secure Coding Team Lead	SEI
Tinnel, Laura	Senior Research Engineer	SRI International
Totah, John	Technical Director in the Office of the CTO	Oracle
van Doorn, Leendert	Corporate Fellow, Corporate VP	AMD
Wagner, Grant	Technical Director, Trusted Systems Research Group	NSA
Weyuker, Elaine	Visiting Scholar, Center for Discrete Mathematics and Theoretical Computer Science & AT&T Labs	Rutgers University
Zurko, Mary Ellen	Security Architect and Strategist	Cisco