Regular Paper

An Architectural Systems Engineering Methodology for Addressing Cyber Security

Jennifer L. Bayuk[1] and Barry M. Horowitz[2,*]

[1]*Stevens Institute of Technology, Hoboken,* NJ 07030
[2]*University of Virginia, Charlottesville,* VA 22901

Received 20 May 2010; Revised 15 September 2010; Accepted 29 October 2010, after one or more revisions
Published online 16 February 2011 in Wiley Online Library (wileyonlinelibrary.com).
DOI 10.1002/sys.20182

ABSTRACT

This paper discusses important shortcomings of current approaches to systems security engineering. The value and limitations of perimeter security designs are examined. An architectural approach to systems security engineering is introduced as a complementary means for strengthening current approaches. Accordingly, this paper outlines a methodology to identify classes of new reusable system security solutions and an architectural framework based on reuse of the patterns of solutions. It also introduces a new methodology for security metrics intended to stimulate critical solution design tradeoff analyses as part of security design reuse considerations. Examples of problems, potential architectural solutions, and corresponding security metrics are provided. © 2011 Wiley Periodicals, Inc. Syst Eng 14: 294–304, 2011

Key words: cybersecurity; security architecture; security metrics; systems security

1. INTRODUCTION

There are an increasing number of opportunities for systems engineers to make important contributions for enhanced protection of systems from cyber attacks. This will require focus on security requirements at the systems level, where *system* means a model of an entity characterized in terms of hierarchical structure, emergent properties, and command and control. It will also require focus on *systems security engineering*, by which we mean the element of system engineering that applies scientific and engineering principles to identify security vulnerabilities and minimize or contain risks associated with these vulnerabilities [DoD, 1995]. Further, it will require the codification of security design principals, which are traditional recommendations for the implementation of security features, into *patterns* that empower reuse.

In the software community, it is recognized that architects, in the course of considering functional aspects of a proposed system, recognize patterns that seem natural to adopt, and thereby benefit from the nonfunctional requirements that the pattern has seen fit to include [Avgeriou and Harrison, 2007]. It is important that systems engineers also embrace the opportunity to provide architectural security patterns and metrics for a variety of reasons. These include, but are not limited to:

1. The risks and threats of cyber attacks that could debilitate systems are substantial and increasing significantly, and therefore require better solutions than are currently employed. Section 2 of this paper provides detail to support this claim.
2. The potential value of creating important reusable system security solutions could be significant, but would require an architectural infrastructure to support the systems engineering community in the selection, integration, and evaluation of solutions, recognizing that

Contract grant sponsor: This work was supported in part by the U.S. Department of Defense under Contract H98230-08-D-0171.

*Author to whom all correspondence should be addressed (e-mail: bh8e@virginia.edu; jennifer.bayuk@stevens.edu).

Systems Engineering Vol 14, No. 3, 2011

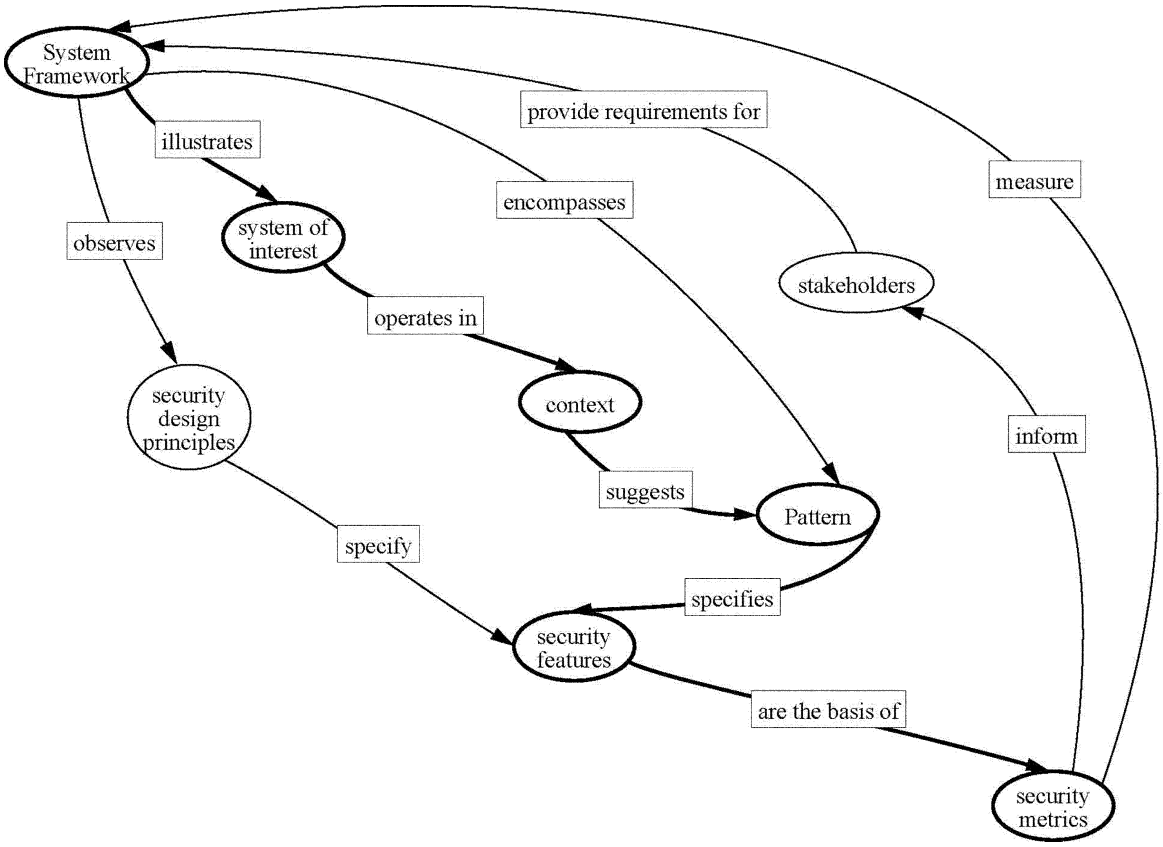

Figure 1. Definition of concepts.

applications would be dependent upon the specific design features and risk profiles of the systems to be protected.

In order to meet this challenge, it will be necessary to:

- Identify classes of new reusable system security solutions
- Provide a security architectural formulation based on reuse of these solutions
- Identify companion security metrics that accompany each new solution and serve to stimulate critical solution design tradeoff analyses as part of reuse considerations.

This paper provides a methodology to support each of the needs cited above. We first identify representative *frameworks*. By framework, we mean an abstraction of the system context with respect to security that can provide the basis for classification of both systems security architecture and associated security solutions [Bayuk et al., 2010]. A framework provides a way to map enterprise asset landscapes to threat landscapes in order to quickly identity system security requirements and test potential solutions. Second, we provide a system security architectural formulation based on reusing security solution design principals as the potential basis for a continuously expanding set of standard system security patterns. By *pattern*, we mean set of architectural artifacts that is useful in implementing a system of a given framework. We reserve the term s*ecurity architecture pattern*, or *security feature*, to refer to the implementation of security design principles that correspond to a framework's architectural context. Finally, we introduce an approach to developing system security metrics based upon the security features and the specific security problems they are intended to solve.

An important aspect of this approach to metrics, as will be elaborated on later in the paper, is that architectural security features are not limited to being add-ons at the perimeter of a system. The security metrics for each architectural feature provide quantitative support for the value of alternative approaches for integrating that feature into the design. Figure 1 is a systemigram that models our usage of these terms.[1]

[1] Explanatory note: A systemigram is read from left to right, top to bottom. Circles contain nouns, which may be objects or concepts. Lines are called threads, which link the nouns. A systemigram describes a system identified in the top left corner succinctly by way of a "mainstay" thread, which connects the system to be defined with its main function or purpose, identified in the bottom right corner. The mainstay is a high level process description that is generally agreed by those who best understand the system. Other threads describe actions taken by the system that, though not central to its purpose, are nevertheless associated with any system so named. A systemigram does not produce a single paragraph of text; many of its threads skirt around its subject in an effort to add dimensions to the definition. See Boardman and Sauser [2008].

This remainder of this paper is divided into five sections. Section 2 discusses the growing systemic risks and threats to systems related to cyber attacks. It also addresses the issues surrounding the use of currently popular security solutions and introduces the more system-specific security solutions as a means for further reducing the effectiveness of cyber attacks on systems. Section 3 introduces the concept of security features as a basis for reusable system security solutions that can be tailored to address system risks specific to a given framework, and introduces an approach for customized security metrics that are directly related to each security feature. Section 4 provides an initial illustrative set of reusable architectural security patterns that, by their nature, could be adapted to address the specific risks of a particular system to be protected. For each of the illustrative architectural patterns presented in Section 4, corresponding examples of potential metrics for system security evaluation are included to highlight the correlation between system-specific risks and the added security being provided by the employed design patterns. Section 5 provides a set of conclusions and recommendations.

2. CYBER THREATS AND CURRENT SECURITY SOLUTIONS

Cyber attacks are a growing problem facing organizations ranging from businesses to governments. These are typically measured by estimates of financial loss; however, any attempt at an exact aggregate dollar amount or other capability loss to represent the increasing risks of cyber attacks is less important than the growing set of causes that contribute to greater risk. These include, though are by no means limited to:

- an increasingly vulnerable supply chain for network communications, electronic circuits, and software [Defense Science Board, 2005; DoD, 2009]
- a steadily rising rate of technically sophisticated and organized adversaries [Menn, 2010]
- a growing tendency to rely on information assurance standards and practices in place of systems engineering approaches to security requirements.

The first two of the above bullets should be uncontroversial to those familiar with current newspaper headlines. Three references so far cited provide ample proof [Menn, 2010; Defense Science Board, 2005; DoD, 2009], and many more are cited in the sections that follow. The third bullet, however, is a major topic of concern. The situation with respect to systems security engineering may not be obvious to those who do not work in the field, and those who do work in the field have no need to draw attention to its obvious flaws. Nevertheless, the obvious certainty of the first bullet above is consequent on lack of adequate methods, processes, and tools with which to address security requirements in today's systems engineering processes [Baldwin, 2009]. A DoD instruction on improving the Information Assurance workforce specifies qualifications for system engineers working in network security at a certification level that can be achieved by taking a week-long course and test on commercially available security technology [DoD, 2005]. A recent, otherwise scholarly and astute textbook on systems engineering, in its only reference to security, says that security is related to attributes that enable a system to comply with regulations and standards [Larsen et al., 2009]. Where security is directly addressed in systems engineering literature, it is addressed as a process by which to ensure security concerns are covered, rather than a product [ISO/IEC, 2002, 2009].

Overreliance on standards and assumptions that stakeholders can articulate security requirements provides false assurance that standards bodies or systems owners understand and know how to address systemic security issues. Because security requirements are generally stated as constraints on systems operation, security in the engineering workplace is increasingly considered an obstacle to operations rather than an enabler of mission assurance [Defense Science Board, 2005]. It is generally recognized that perimeter security is the mainstay of the current cyber security solution space, bringing both important values and serious limitations [Wulf and Jones, 2009]. Even systems security engineers who are critical of standards and best practices write books on the proper use of periphery security, and its corresponding popular mantra, "defense-in-depth" architectures [Anderson, 2008]. The idea is that a security failure at any level may be compensated for by a security measure at the next level. In practice, however, security solutions are access authorization mechanisms that allow users (and attackers) to authenticate (or impersonate) once, and "pass through" multiple levels simultaneously.

Often depicted as a defense-in-depth, bulls-eye target as shown in Figure 2, perimeter security solutions focus on configuring infrastructure to minimize access to system assets at each technology layer of the given system. The engineering benefits of perimeter security solutions include:

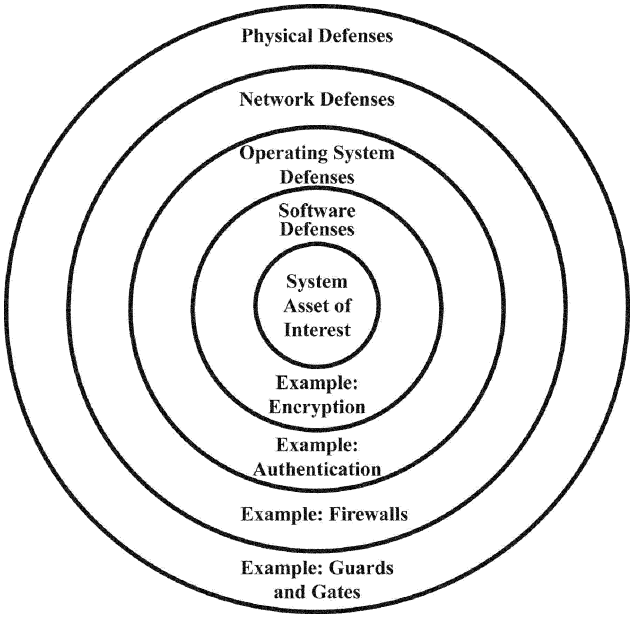

Figure 2. Perimeter defenses.

- They support a policy of noninterference with the development of software functions, which is usually driven by time-to-market pressures, driving designers and developers to avoid the well-known delays that can result from more integrated security solutions.
- As new security needs are identified, they can be added more readily than more integrated security solutions.
- They have been commercialized and commoditized so that the costs and support structures for the solutions have achieved a level of economy of scale that other, more application-specific solutions might not offer.
- They include administrative methods and training as part of the solutions that can be readily adopted by system owners and operators.
- They are supported by standardized sets of best practices [ISO/IEC, 2005a, 2005b; ISF, 2007; ISACA, 2007; Ross et al., 2007].

As a result, the systems engineering and security communities have been able to respond to perceived risks and threats as they arise, by adding perimeter security solutions on a responsive basis.

While this focus on perimeter security has provided some advantages, it also brings with it disadvantages that have become more significant as the cyber threat has evolved. These disadvantages include:

- Systems are frequently not designed to account for the actual cyber attack risks that are inherent in their functional and technical designs. For example, embedded software is becoming a more and more important element of physical systems so as to permit remote control through networks; controls that may become accessible to cyber attackers [Weiss, 2010].
- The use of customized application-specific solutions has not been an important aspect of the cyber security solution space, greatly reducing the toolsets that can potentially address cyber attack risk reduction [Wulf and Jones, 2009].
- The cyber security workforce (e.g., designers, operators, administrators) frequently do not know enough about the "business" risks of the systems that their security components support, while the people who know the "business" risks frequently do not have the needed detailed technical knowledge about solutions. This regularly leads to misappropriated risk reduction.
- Commercially available perimeter solutions are well known and available for study, such that cyber attackers can discover and exploit vulnerabilities, and can reuse their solutions across the base of systems that use common products. In the most perverse cases, cyber attackers can participate in the supply chains that produce the off-the-shelf software or hardware depended upon to be secure [Markoff, 2008].
- Systems are not designed to anticipate possible new security threats and solutions, resulting in unnecessary difficulties in integration of new solutions as they emerge.
- Individual systems that are embedded within a system-of-systems enterprise architecture are typically not designed to withstand attacks through solutions that exploit the relationships among the peer systems (e.g., depend upon members of the system-of-system consortium to serve as redundant elements in the event of a denial of service attack on a specific system, rely on members of the consortium to inform each other regarding threat information).
- In view of the growing threat of successful attacks, the disadvantages presented above are likely to become much more important over time. There are three principal reasons:
- As described in Wilson [2009], Adair et al. [2010], Acohido and Swartz [2008], and Clarke and Knake [2010], nature of the threats is transitioning from being dominated by individuals initiating attacks to organized cyber attack criminal groups and nation-state sponsored attack groups. This has increased both the level of investment and the sophistication of exploits that are available to attackers.
- Defensive efforts are at a disadvantage to offensive activities, since defenses must consider protecting against all credible system attacks that can create important consequences, while offenses are at greater liberty to select the specific attacks that they wish to discharge. This difference puts the onus on defenses to develop well thought out techniques for selectively deciding on investments in security solutions (Michael Howard describes this situation as "the attacker's advantage and the defender's dilemma" in Howard [2002]).
- The technology industry is moving toward more integrated components in efforts to reduce costs in areas such as hardware, administration, data management, and communications. These technology advances then provide the opportunity for third-party organizations to provide infrastructure for computing as a service. Cloud computing is a major initiative for achieving these new efficiencies [Mather, Kumaraswamy, and Latif, 2009]. Integration further results in greater concentration of computational functions, which then create new and greater risks related to cyber attacks.

The perimeter defense situation described above points to the fact that current security solutions are most frequently developed through a responsive engineering approach that can be characterized as bottom-up; starting with the components and working to get the most security they can offer, on an as-needed basis. The premise for this paper is that there are:

- limitations to a responsive approach
- important trends in the threat landscape that further enforce these limitations
- increasingly integrated systems
- few available security architecture patterns
- opportunities for framework-specific security solutions.

Taken together, these observations are sufficient to warrant the use of a top-down, risk-based systems security engineer-

ing approach that is complementary to currently used perimeter security approaches, an approach that develops strategies regarding cyber security in the broadest sense, starting from the time a new system is developed and continuing through its entire life cycle. While significant attention has been paid to best practices for dealing with the bottom-up approach for engineering security design principles, the systems engineering community has yet to develop a corresponding framework for a top-down approach to address cyber security.

Figure 3 is a standard textbook model of the top-down engineering design and integration process (the diagram is based on Buede [2009]). It is annotated with the point in the systems engineering process where security should be considered, irrespective of whether it is an explicit stakeholder requirement. Performance and functional specification and validation plans are in some sense meaningless without a concept of system survivability. Security design principles that should be considered for incorporation into "design-to" specifications include a full suite of well-known security features such as nonrepudiation, fail-safe default, economy of mechanism, transparency of design, and ease of authorized use [Bishop, 2003].

This approach to incorporating security into the systems engineering process should be considered in the context of the system operational concept and associated emergent threat landscape, in full recognition that both may change in the course of the design process. It requires consideration of the systems' support environment and an expansion of system boundary into the people, processes and technology that maintain and update the components during the operational phase of the lifecycle [Mulokey, 2009]. The approach should provide the basis for the systems engineering community to integrate information assurance and cyber security as regularized functions in the broad scope of their overall efforts, dealt with in a comparable manner as systems engineers are expected to treat other system attributes, such as reliability, maintainability, availability, supportability, and so on.

3. SECURITY ARCHITECTURE AND METRICS

Expertise in any field may be associated with pattern recognition, and systems architecture is no exception. Patterns can succinctly provide relatively large amounts of information and understanding that can be directly tied to a target system architecture. As described by Cloutier and Verma, patterns emerge from the engineering community as successful precedents, reflecting consensus among engineers [Cloutier and

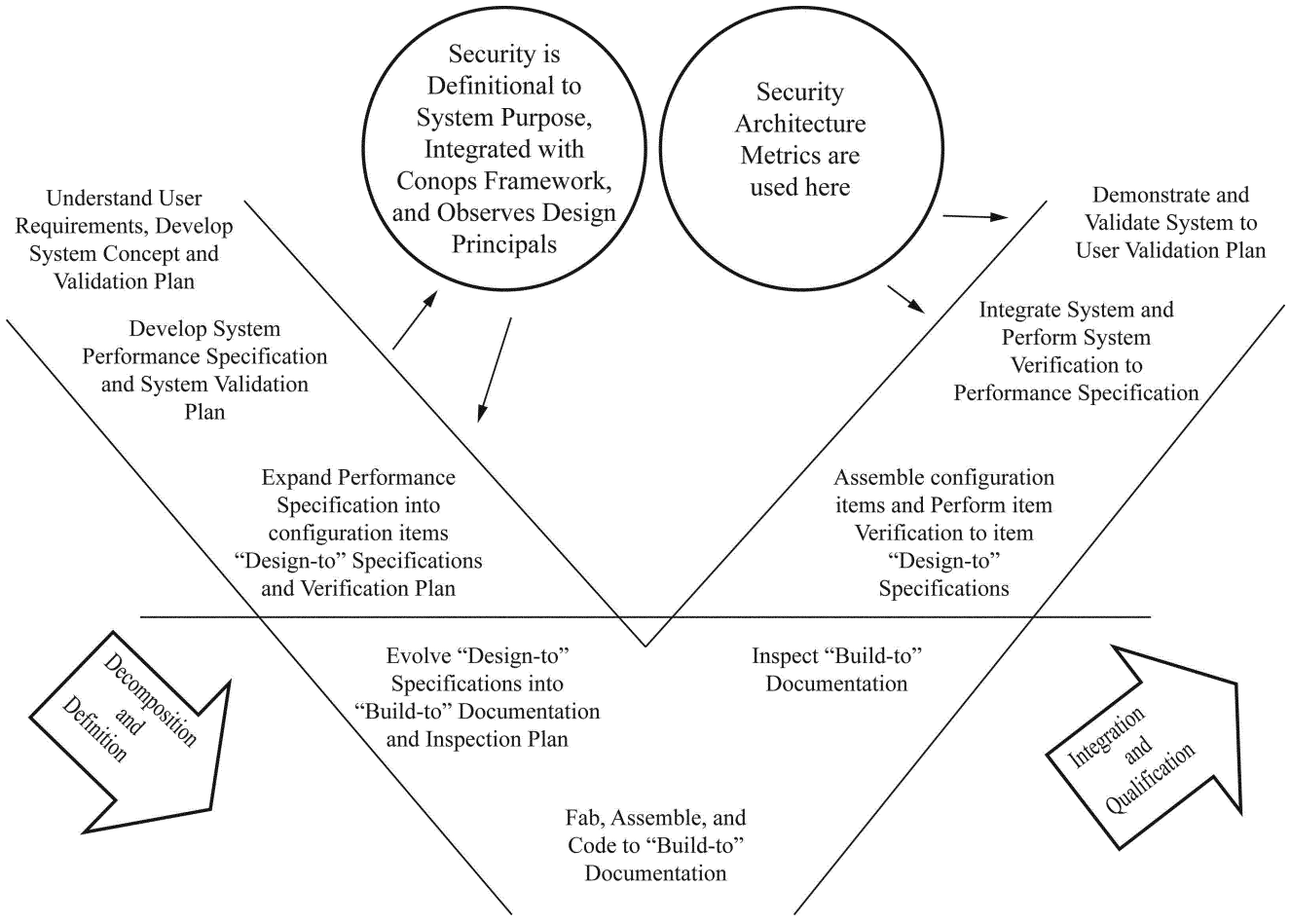

Figure 3. Systems engineering process.

Verma, 2006]. When documented, they provide a template by which to map an engineering problem to a design solution. Communities with similar functional requirements and similar interests in security (these include all owners of Emergency Support Functions and Key Resources as identified by the US Department of Homeland Security [FEMA, 2008] as well as military and intelligence communities) may be encouraged to document and share architectural patterns that they recognize as successful in addressing security requirements.

The perimeter defense strategy and the associated defense-in-depth engineering approach is one example of a security pattern. It qualifies as a pattern because engineers commonly apply it as a security solution to any framework that looks like an electronic process in need of security. Applying the pattern, as opposed to customizing a system security engineering process, provides cost and implementation time reductions, and relieves the engineer of the burden of invention. The computer security literature includes many defense-in-depth perimeter patterns that realize security features using encryption, authentication, and network segmentation technology components that are well established in commercial security products [Schumacher et al., 2006]. Associated metrics include a combination of coverage and control metrics designed to ensure that known threats and vulnerabilities to network inventory are countered with appropriate technical configuration [Jaquith, 2007]. These correspond to "build-to" documentation in the diagram of Figure 3.

However, as is evident from the ease with which adversaries penetrate today's security technology, systems engineers rely too heavily on these network-centric perimeter patterns, and more diversified architectural security solutions, or patterns, are needed. Metrics that correspond to secure architecture choices as opposed to technical verification of control implementation are needed as well. These would be the security architecture metrics above the horizontal line in Figure 3.

This recognition has been a significant focus area of the International Council on Systems Engineering (INCOSE) security working group [Dove and Shirey, 2010]. INCOSE patterns are described contextually using the following characteristics:

Name: Descriptive name for the pattern.
Context: Framework to which the pattern applies.
Problem: Description of the problem.
Forces: Tradeoffs, value contradictions, key dynamics of tension and balance, constraints.
Solution: Description of the solution.
Graphic: A depiction of response dynamics.
Agility: Evidence of characteristics that qualify the pattern as agile and adaptable.
Examples: Referenced cases where the pattern is employed.

An example of using the INCOSE pattern template to describe the perimeter security situation would be:

Name: eCommerce.
Context: Online retail shopping.
Problem: Multiple vendors must be able to securely process credit card information without exposing shoppers to identity theft.
Forces: Authorized users are required to enter these data, system processors must communicate these data to third parties in order to execute transactions, and these third parties need to see the data in clear text.
Solution: Keep sensitive data in an encrypted database, and only allow it to be accessed via applications that provide authentication, and restrict those application communication to network links that are encrypted.
Graphic: See Figure 2.
Agility: Administrative configuration utilities allow changes in user communities, data entitlements, and firewall rules.
Examples: Payment Card Industry (PCI) Data Security Standard, Version 1.2 [PCI, 2008].

Note that simply to be able to describe a pattern in template format does not make the pattern a successful precedent. Many would argue that this defense- in-depth pattern should not be applied to eCommerce environment because the sheer magnitude of identity theft cases shows this pattern to be a failure. The patterns that the INCOSE security working group is suggesting also do not meet the Cloutier and Verma [2006] criteria for engineering patterns because they are being created rather than emerging. They have not yet been implemented, much less vetted. However, it is possible to use the INCOSE security pattern template to show that there are actually emergent security patterns in today's architecture that may be described in the context of a security problem with a given framework and corresponding security solution.

Security architecture patterns can also provide a basis for defining system security metrics to be used for security validation and verification; metrics that are directly correlated to the needs of the system being secured as well as the specific security design principles employed. These framework-specific metrics would provide independent security assessors who are not intimately familiar with the system under evaluation with system-specific security criteria. As in traditional security metrics, these criteria would be related to the verification that mechanisms were implemented according to design. However, unlike the case of traditional security metrics, the framework would also provide system-specific functional criteria for measurements as part of the system specification. In the next section, we present examples of systemic security issues that may be addressed with framework-based security solutions and associated metrics at a level associated with systemic security features.

4. SECURITY DESIGN PATTERNS AND METRICS

While there has been little implementation of system-specific security solutions, there are a variety of solutions classes that could be developed if the risks warrant it and the economics of development are suitable. As discussed in Sections 2 and 3, the risks of cyber attacks are increasing, and the use of design patterns offers a potential approach to reducing the costs of implementing system specific solutions. A recently published DoD systems security engineering research roadmap identified several example security frameworks: Critical

Program Information Protection, System of Systems, Configuration Hopping, Continuity of Communications, Data Continuity Checking, Denial and Deception, Shared Command Information Sharing, and Physical Security [Bayuk et al., 2010]. The concept of a framework allows for countless others. Given the discussion of Section 3, potential solutions to system security requirements for a given framework would be considered security architecture patterns. Once a design pattern is identified, it or elements of it may turn out to be effectively applicable to all systems in the framework for which it was developed, or it may have broader applicability due to similarities between frameworks.

This section addresses the reusability of system-specific solutions through the employment of security architecture patterns. Reusable patterns may be closely related to, and dependent upon, the specific functionality and design of a system of interest. Security patterns derive utility by drawing on the consensus of engineers engaged in building systems characterized by a certain framework. The two frameworks below illustrate by example how the utility of a security feature is identified and corresponding security metrics derived for the given type of system.

4.1. Data Continuity Checking

Important surveillance system designs integrate pipeline computing processes. These typically start with a data collection function, and progress through a pipeline of data processing, including computational processes such as object detection, object location tracking, integration of correlated reports from multiple data collection sources, object identification, and object presentation to operators responsible for managing or responding to observations.

There are numerous examples of systems that include such capabilities. They include air traffic control systems collecting radar surveillance reports to support air traffic management functions, and military warning systems collecting infrared and radar reports for alerting the military command to an Intercontinental Ballistic Missile (ICBM) attack so as to enable timely military responses. In practice, the architecture for executing such processes varies, ranging from centralized to highly distributed computing configurations.

As described in Section 2, security features that should be incorporated into "design-to" specifications include security principles, such as access controls, fail-safe defaults, economies of mechanism, transparency of design, and ease of authorized use. These design principles may not have been formally identified by engineers in the air traffic or ballistic missile systems, but, over the decades of implementing many types of data continuity systems, consensus has been built that allows us to identify security features that are common to successful implementations of systems of this type. For any such configuration, consider the possibility of a persistent threat (present in either hardware or software) that could either:

a. prevent data from being properly processed in order to avoid operator observation of a specific object(s) or
b. create artificial data as a decoy to attract operator attention away from other data that should be acted upon.

The significance of this example was observed in 1980, when the US ICBM warning system indicated to its operators that a full-scale nuclear attack had been initiated against the United States [US Comptroller General, 1981]. The strategic command and control system, which included the participation of the President, responded with a set of actions that included immediate dispersal of nuclear-armed bombers into holding patterns to await follow-on orders (so as to avoid their destruction on the ground). Of course, the event was a false alarm, later discovered to be caused by malfunctioning hardware.

After-the-fact analysis raises the question as to why the warning system wasn't designed to recognize a condition in which there were data indicating an attack on the screens being observed by operators, while at the same time there were no missile-related data being received from the system's sensors. Such an arrangement is referred to as data continuity checking, and the only answer to this question is that the systems engineering function did not recognize this as a needed security feature for the warning system. While this event was stimulated by malfunctioning hardware, it could just as well have been a Trojan horse inserted through the supply chain and controlled, for example, by an insider.

This example points to the opportunity for systems engineers to develop continuity checking software agents that help to assure data integrity. For example, a "**data continuity agent**" for decision support system applications would collect and analyze metadata from selected components in the pipeline to help assure that an externally injected change has not occurred through the actions of a Trojan horse. The design of such a data continuity agent requires:

1. **Development of a taxonomy for associating data elements to decisions in a manner that helps system users to relate externally forced changes in decision support data to potential critical decision errors.** Examples would include:
   - identification of single data elements that can change a critical decision regarding a single object (e.g., "friend" or "foe" designation);
   - identification of single data elements that can change a decision regarding pairs or groups of objects (e.g., change of the observed altitude for a radar observation of an aircraft so as to create what appears as a possible collision opportunity with another aircraft);
   - elimination of data that the surveillance system is designed to deliver to responders; or
   - creation of false data that diverts the efforts of responders.
2. **Development of a user interface to the data continuity agent for:**
   - designating the metadata that will be available to the agent from the various service components in the system;
   - the comparisons that the agent is required to make as a basis for recognizing a discontinuity;

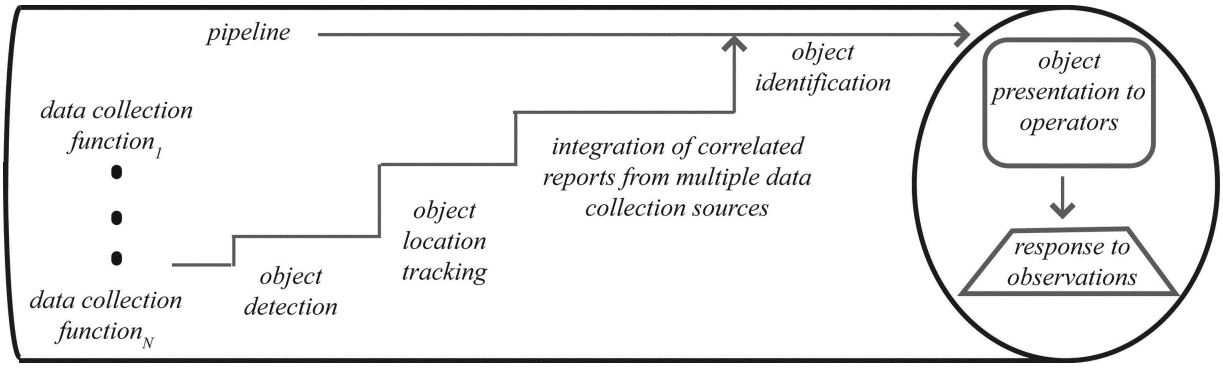

Figure 4. Data continuity pattern.

- the time lines to be used for making comparisons; and
- the interface that the agent should use for reporting discontinuities.

These design issues point to the system engineers as the logical source of the solution to security problems in this systems context. It thus qualifies as a security pattern.

The reusability of security features instantiating data continuity agents would be of interest, and it is possible that such agents could be designed for reuse across a defined set of system specifications. The data continuity pattern in the context of an air defense system would look like this:

Name: Data Continuity.
Context: Situation in which data are collected from multiple sources and processed through a sequence of distinct computational processes that independently detect, locate, track, correlate, and present to operators responsible for managing or responding to observations.
Problem: As operators are reacting to data rather than actual objects, any point in the computational pipeline between physical object sensors and screen presentation is a potential vulnerability if exposed to threat.
Forces: Limitations in the resources available for physical object detection, and the fact that they must share an environment with a tracked object which may be hostile, exposure of data pathways, reliance on economies of scale by commercially available processors. The system must facilitate repudiation for hostile activities designed to enact threats while maintaining nonrepudiation for authorized information flow.
Solution: Taxonomy and user interface as in the requirements for the design of such a data continuity agent in the numbered list 1 and 2, above.
Graphic: See Figure 4.
Agility: Anticipation of technology advances via modular components allowing multiple alternative interfaces and computer platforms.
Examples: Air Defense and Response system.

Assuming that the design of the system followed the guidelines recommended in the solution and agility criteria, corresponding metrics would be devised that would provide a clear indication of the strength of the security built into the design. As indicated in the figure, it should, for example, also be possible to relate the security provided in this framework to the false alarm rates caused by discontinuities from nonmalicious sources such as (a) possible tracking errors, (b) sensor signal-to-noise ratios, (c) data continuity quantization levels, (d) data update rates, (e) frequency for data continuity checking, and (f) the type and number of information delivery alternatives available to the end user/operator. The security architecture pattern for the data continuity framework could include metrics that relate to each of the individual factors cited above, as well as metrics that relate to the group as a whole. The integrated metrics could include ordinal and cardinal indications, with the security assessor selecting the most appropriate for the system and risks under evaluation.

4.2. Resiliency of Communications

As in the continuity of data example, security design principles have been tacitly observed by qualified engineers in the framework of communications systems. Consensus has been reached that allows us to identify security features that are common to successful implementations of systems of this type. This example is related to deterring and responding to attacks intended to disrupt communications between geographically separate components of a system.

A well-recognized denial of service threat to systems involves physical disruption of communications that connect separated elements in a system. A frequently used security architecture pattern to respond to this kind of threat is the provisioning of redundant communications sources, such as receiving redundant landline communications through alternative routing paths that are spatially separated, making acts of sabotage more difficult to carry out without being discovered during the attack. This pattern can be expanded to include multiple modes of communication such as supplementing landline communication with multiple radio communications systems for providing continuity of communications. However, for systems where the normal bandwidth requirements for data transfers exceed the available bandwidth of a radio system used to provide continuity, functional components of the system must be modified to either operate in modes tolerant of larger delays in receiving data, or in modes that can

acceptably work with reduced data content in order to keep communications delays within the normal system specifications. This security architecture pattern would provide the system adjustments necessary for either accommodating greater delays or reducing the amount of data transmitted.

For example, communications to support military warning systems are subject to sabotage and electronic warfare. Data delays must be kept to a minimum because warning is a precursor to what can be time-critical responses. As a result, for applying this system security architecture pattern, data sent over low bandwidth radio systems for providing continuity of long-range communications must be compressed to avoid unwanted delays. As a particular example, a warning system might normally receive relevant remote sensor surveillance information over dedicated landline telecommunications system at 9600 bits per second. For that same system, it may be desirable to use an HF radio system as a redundant source, but only capable of delivering the data at 2400 bits per second. The system designer may choose to delay the data by a factor of 4, or alternatively the data can be compressed by a factor of 4. Assuming that added delay is not acceptable, compression is required. One method of compression could be to change data quantization levels. For example, for a warning system that receives communications regarding locations of sources of attack, instead of sending locations with 0.1 mile precision, the communicated data can locate the sources with less precision (e.g., 1 mile precision), thereby reducing the number of bits required for transmissions. Further assume that the system designers would like even greater security regarding disruption of communications and would like to add a lower frequency communications system that supports only 100 bits per second of communication in addition to the HF system. This large a reduction from the normal 9600 bits per second sent by landline could warrant the design of a new mode of system operation that uses summary reports rather than individual location reports (e.g., "there are ***3*** sources of attack coming from the ***northern sector*** of enemy locations") with only the quantity and sector description being the communicated data. While the specifics of this example would be highly dependent on the particulars of the framework that was being secured, the general security architecture pattern could likely be used on a variety of systems requiring similar security solutions.

The map from this context, problem, and solution description to a security architecture pattern is straightforward, and therefore will be omitted as unnecessary for the reader to peruse. Security metrics corresponding to this architecture pattern would include the number of physically different communications paths, the number of logically distinct communications protocols, and information-theoretic statistics that demonstrated ability to provide mission-critical information when operating at reduced capacity.

5. CONCLUSIONS AND RECOMMENDATIONS

An unintended consequence of using off-the-shelf perimeter security products has been the loss of system-specific security innovations that should be investigated and explored by the systems engineering community. This paper has highlighted the growing risks of cyber attacks and inadequacy of current popular perimeter solutions that have to date been the main methods for engineering computer security. We have presented an architectural approach for enabling systems engineers to efficiently provide more application-sensitive and system-specific risk-focused solutions that are reusable; that is, to rely upon the use of standard design patterns for addressing regularly encountered system security problems. This paper provides descriptions of two reusable design pattern examples that provide a basis for further work related to this concept.

In addition, the paper highlights the concept of developing pattern-specific metrics that would capture the knowledge of the pattern creators regarding the relationships between the particular instantiation of the pattern in its application and the level of security that would be provided. This approach to metrics would allow security solutions to be evaluated more effectively by security assessors who do not have complete knowledge of the system being secured.

While the suggested architectural approach may not offer all of the efficiencies associated with reusable off-the-shelf software, it would offer a new and potentially valuable set of solutions in a manner that would offer its own efficiencies and would add new security capabilities that are not offered by perimeter security solutions. Through the use of an architectural methodology that reuses design patterns, system engineers can potentially create solutions that are both efficient from a cost and schedule viewpoint, and that also provide security in a manner that is unique to the systems being protected. As an important added benefit, the unique implementations of design patterns would be problematic to attackers who depend upon developing exploits that can penetrate off-the-shelf perimeter solutions and can often reuse those exploits across the base of systems who use the same products for their security.

REFERENCES

B. Acohido and J. Swartz, Zero day threat, Sterling, New York, 2008.

S. Adair, R. Deibert, R. Rohozinski, and G. Walton, Shadows in the cloud: Investigating cyber espionage 2.0, A joint report of the Information Warfare Monitor and Shadowserver Foundation, Toronto, 2010.

R. Anderson, Security engineering, 2nd edition, Wiley, Hoboken, NJ, 2008.

P. Avgeriou and N. Harrison, Pattern-driven architectural partitioning, balancing functional and non-functional requirements, Second Int Conf Digital Telecommun, IEEE, 2007.

K. Baldwin, Systems security engineering: A critical discipline of systems engineering, INCOSE INSIGHT 12 (2009), 11–13.

J. Bayuk, D. Barnabe, J. Goodnight, D. Hamilton, B. Horowitz, C. Newman, and S. Tarchalski, Systems security engineering roadmap, final technical report, Systems Engineering Research Center (www.sercuarc.org), Stevens Institute of Technology, Hoboken, NJ, 2010.

M. Bishop, Computer security, art and science, Pearson Education, Upper Saddle River, NJ, 2003.

J. Boardman and B. Sauser, Systems thinking: Coping with 21st century problems, Taylor & Francis, New York, 2008.

D.M. Buede, The engineering design of systems, models and methods, Wiley, Hoboken, NJ, 2009.

R.A. Clarke and R.K. Knake, Cyberwar, HarperCollins, New York, 2010.

R. Cloutier and D. Verma, Applying pattern concepts to systems (enterprise) architecture, J Enterprise Architects 2(2) (2006), 138–154.

Defense Science Board, High performance microchip supply, Department of Defense, Washington, DC, 2005.

DoD, MIL-HDBK-1785, System security engineering program management requirements, US Department of Defense, Washington, DC, 1995.

DoD, Information assurance workforce improvement program, DoD 8570.01-M, US Department of Defense, Washington, DC, 2005.

DoD, The Under Secretary of Defense for Acquisition Technology and Logistics and The Assistant Secretary of Defense for Networks and Information Integration DoD Chief Information Officer, Report on trusted defense systems, Department of Defense, Washington, DC, 2009.

R. Dove and L. Shirey, On discovery and display of agile security patterns, Conf Syst Eng Res, Stevens Institute of Technology, Hoboken, NJ, 2010.

FEMA, National response framework: Overview, http://www.fema.gov/emergency/nrf/, US Department of Homeland Security, Washington, DC, 2008.

M. Howard, Writing secure code, 2nd edition, Microsoft Press, Seattle, WA, 2002.

ISACA (Information Systems Audit and Control Association), Control objectives for information technology (cobit), IT Governance Institute, Rolling Meadows, IL, 2007.

ISF (Information Security Forum), The standard of good practice for information security, London, 2007.

ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), Information technology—systems security engineering—capability maturity model, sse-cmm, ISO/IEC 28127, Geneva, Switzerland, 2002.

ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), Information technology—security techniques—information security management systems—requirements, ISO/IEC 27001, International Standards Organization, Geneva, Switzerland, 2005a.

ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), Information technology—security techniques—code of practice for information security management, ISO/IEC 27002, Geneva, Switzerland, 2005b.

ISO/IEC (International Organization for Standardization/International Electrotechnical Commission), Systems and software engineering—systems and software assurance—part 2: Assurance case, ISO/IEC 15026, Geneva, Switzerland, 2009.

A. Jaquith, Security metrics, Pearson Education, Upper Saddle River, NJ, 2007. Table 3-3.

W. Larson, D. Kirkpatrick, J. J. Sellers, D. Thomas, and D. Verma, Applied space systems engineering, McGraw Hill, New York, 2009.

J. Markoff, FBI says the military had bogus computer gear, The New York Times, May 9, 2008.

T. Mather, S. Kumaraswamy, and S. Latif, Cloud security and privacy: An enterprise perspective on risks and compliance, O'Reilly, Sebastopol, CA, 2009.

J. Menn, Fatal system error, Perseus Books Group, Jackson, TN, 2010.

W.P. Mulokey, Using the U.S. Department of Defense architecture framework to build security into the lifecycle, INCOSE Insight, 12 (2009), 27–29.

PCI, Payment Card Industry (PCI) Security Standards Council, Payment card industry (pci) data security standard, version 1.2, Wakefield, MA, 2008.

R. Ross, S. Katzke, A. Johnson, M. Swanson, G. Stoneburner, and G. Rogers, Recommended security controls for federal information systems, sp 800-53 rev 2, National Institute of Standards and Technology, Gaithersburg, MD, 2007.

M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, Security patterns, integrating security and systems engineering, Wiley, Hoboken, NJ, 2006.

US Comptroller General, Report to Chairman, Committee on Government Operations, House of Representatives of the United States, Washington, DC, May 1981.

J. Weiss, Protecting industrial control systems from electronic threats, Momentum Press, Williston, VT, 2010.

C. Wilson, "Cybercrime," Cyberpower and national security, F.D. Kramer, S.H. Starr, and L. Wentz (Editors), Potomac Books, Dulles, VA, 2009, Chap. 18.

W. Wulf and A. Jones, Reflections on cyber security, Sci Mag 326 (2009), 943–944.

Jennifer L. Bayuk is the Cybersecurity Program Director of the School of Systems and Enterprises at Stevens Institute of Technology. She has been a Wall Street Chief Information Security Officer (1998–2008), a Manager of Information Systems Internal Audit (1997–1998), a Price Waterhouse Security Principal Consultant and Auditor (1995–1997), and a Security Software Engineer at AT&T Bell Laboratories (1990–1995). Bayuk frequently publishes and speaks on IT Governance, Information Security, and Technology Audit topics. She is the author of *Stepping through the IS Audit*, 2nd Edition (ISACA, 2004), *Stepping through the InfoSec Program* (ISACA, 2007), and *Enterprise Security for the Executive* (Praeger, 2010). She also has coedited a collection of works on enterprise information security and privacy. Bayuk is a Certified Information Security Manager, a Certified Information Systems Security Professional, a Certified Information Security Auditor, and Certified in the Governance of Enterprise IT (CISM, CISSP, CISA, and CGEIT), as well as a member of INCOSE, IEEE, and ACM.

Barry M. Horowitz is the Walter Munster Professor and Department Chair of Systems and Information Engineering of the School of Engineering and Applied Sciences at the University of Virginia. Prior to joining the university in 2001 he was Founder, Chairman, and CEO of Concept Five Technologies, an e-business systems development company (1996–2001) and from 1969 to 1996 served in a variety of positions at the Mitre Corporation including CEO. He was elected as a member of the National Academy of Engineering in 1996 and was awarded the United States Air Force Exceptional Service Award in 1992.