Carnegie Mellon University
Software Engineering Institute

P21-071: Challenges in Building and Implementing an Effective Cybersecurity Strategy

Carol Woody, Ph.D.
412-770-5133
cwoody@cert.org

Rita Creel
703-247-1378
rc@cert.org

May 2021

Abstract

Today's missions rely on highly integrated and complex technology that must be protected from a wide range of adversaries. This technology must also operate in a very dynamic and contested environment.

A cybersecurity strategy is a critical element in defining how each technology component and its composition will have sufficient security to address a mission. This strategy requires planning, design, monitoring, and enforcing considerations of cybersecurity at all levels of process, practice, and technology. It is necessary to consider compliance mandates for an authority to operate and achieve good cybersecurity hygiene. However, these steps alone are not sufficient to ensure each component is adequately secure since the capabilities of technology are continually expanding along with attackers' abilities.

Effective cybersecurity requires applying engineering rigor to the process of defining requirements and preparing the technology to handle the operational environment where it will ultimately reside. In this paper, we describe the challenges that acquisition programs face and the ways they are attempting to address gaps as the need increases for improved cybersecurity.

Current Approaches Are Too Limited

Most programs currently address cybersecurity by relying on a small cadre of security experts that function as a silo of excellence that separates lifecycle and acquisition activities into independently managed teams. This small group typically has limited systems and software knowledge and focuses on the ways that compliance mandates must be addressed. However, implementing an effective cybersecurity strategy needed for today's missions requires extensive collaboration among all lifecycle participants to ensure each is carrying out their portion of the strategy and moving the product toward an effective cybersecurity outcome. Cybersecurity engineering expertise will be needed to create and field systems that exhibit desired cybersecurity behaviors.

An effective cybersecurity strategy must define how acquisition and development lifecycle activities are performed so that the resulting system can be fielded with low risk. The roles and responsibilities of cybersecurity experts during acquisition and development include owning and leading the strategy, which focus on searching for and removing the following:

- gaps that allow unexpected and unwanted interactions among components and external systems
- weaknesses in the design and code that make the system vulnerable
- opportunities for processes to bypass controls, allowing attackers to impact mission success

Cybersecurity engineering should augment the design, considering what could go wrong and monitoring system creation and verification to confirm that the system is prepared for potential operational misuse and abuse.

What Is an Effective Cybersecurity Strategy?

A cybersecurity strategy is a critical element used to ensure that each component and its composition have sufficient security to address a mission. This strategy does not occur without planning, design, monitoring, and enforcing considerations of cybersecurity at all technology levels. Compliance mandates that grant an authority to operate and good cybersecurity hygiene must be considered. However, these steps alone cannot ensure the composition is sufficiently secure. Cybersecurity responsibilities touch on every aspect of the lifecycle. An effective approach requires a high level of collaboration; cybersecurity engineering must cross all activities, and that collaboration must not be assumed.

The owners of the cybersecurity strategy are responsible for defining how a system's cybersecurity performs to meet the system's mission, even when under attack. These responsibilities include activities that achieve the following:[1]

- Plan and design trusted relationships.
- Negotiate appropriate security requirements to ensure confidentiality, integrity, and availability with sufficient monitoring in systems and software to identify problems.
- Plan and design a system with sufficient resiliency to be able to recognize, resist, and recover from attacks.
- Plan for operational security under all circumstances, including designed-in methods of denying critical information to an adversary to avoid or minimize mission impact.

---

1 Key elements of an effective cybersecurity strategy were introduced in a webcast delivered in November 2020 (Woody & Creel, 2020).

- Evaluate alternatives to select an acquisition option with an acceptable level cybersecurity risk.

Integrating cybersecurity considerations into the early segments of a lifecycle affects acquisition as well as system and software engineering. Program management is ultimately responsible for cybersecurity;[2] however, if management lacks that specific expertise, it must bring in resources with specialized knowledge and operational expertise, which we describe as *cybersecurity engineering*.

The right people to augment the program office must understand systems, software, and cybersecurity. These people must be sufficiently committed to this task to have impact; just attending a few meetings every now and then will not make a difference. Existing security verification tools and practices can help, but they will not be completely sufficient since they focus primarily on code. Flaws and weaknesses can be introduced early in the lifecycle even before code is introduced.

The full range of participants in developing the strategy must build a shared understanding of what is to be accomplished and who will handle which pieces. For some, this shared understanding means building a shared vocabulary since each discipline uses similar terms to mean very different things. Participants must also have a shared understanding of the mission and how security risk can impact mission success.

Cybersecurity engineering should focus on the following six key areas. Although these areas are critical for building technology to operate in today's highly contested environments,[3] they are typically given insufficient consideration:

- determining risk
- defining and monitoring system and component interactions
- evaluating trusted dependencies
- anticipating and planning responses to attacks
- coordinating security throughout the lifecycle
- measuring to improve cybersecurity

More details about defining the cybersecurity strategy are available in the paper ***Building a Cybersecurity Strategy*** by Carol Woody and Robert Ellison. It was published in the Special Issue on Rigor and Inter-Disciplinary Communication of the *Journal of Systemics, Cybernetics and Informatics* (Woody & Ellison, 2020a).

Building a cybersecurity strategy is challenging, and applying it to a system under acquisition and development is even more so. We worked with a broad range of acquisitions to tackle this important need, but changing the existing patterns of cybersecurity neglect is extremely difficult.

---

2 See the DoD memorandum, *Outline and Guidance for Acquisition Program's Cybersecurity Strategies* (Department of Defense, 2015).

3 See the book chapter, "Principles and Measurement Models for Software Assurance" (Mead et al., 2013).

What Makes Effective Cybersecurity Engineering Challenging?

In the remainder of this paper, we explore some of the challenges of changing existing patterns and improving cybersecurity experts' involvement in acquisition development. We also share effective responses we have seen to date to address the six key areas of cybersecurity engineering.

Challenges in Determining Risk

Perceptions of risk drive assurance decisions, and the lack of cybersecurity expertise in acquisition and development risk analysis has led to poor assurance choices. Involving participants who have knowledge about successful attacks and how threats can impact a system's operational mission can be very useful in the decision-making steps for determining appropriate prioritization. However, to effectively communicate cybersecurity issues to leadership and systems and software engineers in the program office, these participants must have more than a rudimentary understanding of the system and software acquisition and development lifecycle.

Risk considerations must start with good cybersecurity requirements, but simply providing a list of standards and policies—an approach too frequently used by programs—does not ensure that risks to the specific mission and technology are addressed. New versions of standards and policies are now appearing so fast that programs have little time to respond to them; they then continue working with outdated versions until contract modifications can be made. We have also seen programs fail to establish even minimum cybersecurity or compliance requirements in the contract and the vendor plans to deliver a system that will not be implementable without contract modifications and further development. When developing good cybersecurity requirements, participants must do the following:

1. understand how the many layers of technology to be implemented to address a mission can be compromised
2. define requirements that enable consideration of the broad and ever-increasing threat environment
3. clearly define how success will be verified

Requirements for many aspects of a system contribute to cybersecurity. The structure of these requirements affects how the vendor will respond. When requirements for anti-tamper, supply chain, information security, software assurance, cybersecurity, and safety controls are developed by independent teams that do not collaborate, the contractor cannot clearly propose a system that meets what could be conflicting needs. The vendor will make choices based on their perspective which will likely differ from the intent of the program. Adjustments must be made later in the lifecycle as time and funding permit, and the results will vary widely depending on the knowledge of the decision makers involved at each subsequent stage. Security teams are usually too small to support all of the interactions needed for this approach to cybersecurity risk decision making.

In many cases, the cybersecurity risks and potential mission impacts of choices made in design and engineering are not well understood by system and software engineers who (1) were not trained in cybersecurity and (2) do not understand the capabilities of attackers in the contested environment where the systems are fielded. The program too frequently accepts risks without understanding the mission impact.

Challenges for Defining and Monitoring System and Component Interactions

In systems design, we see engineers rush to decompose the system into technology components and delegate requirements to these various pieces. Interface protocols and application programming interfaces (APIs) are defined to provide mechanisms for data sharing among components. Cybersecurity controls are selected from a wide array of sources and sprinkled like "fairy dust" among the components. However, nowhere is there a coherent plan for how the overall system cybersecurity will operate across the attack surface of the system. Views of the hardware, network interfaces, software interfaces, and mission capabilities are provided, but the operational aspects of cybersecurity are fragmented.

To address this void, programs must define how the technology will support mission execution and evaluate how information flows and data exchanges can be attacked. To conduct a cyber-threat analysis of a system, we applied the Security Engineering Risk Analysis (SERA) Method (Alberts et al., 2014). The SERA Method defines a scenario-based approach for analyzing complex security risks in software-reliant systems and systems of systems. The SERA Method incorporates a variety of models that can be analyzed at any point in the lifecycle to (1) identify security threats and vulnerabilities and (2) construct security risk scenarios. An organization can use those scenarios to evaluate whether planned controls will be sufficient and focus its limited resources on enhancing how to meet the needs of mitigating the most significant remaining security risks.

For the most part, we see programs failing to assemble the information they have in ways that enable their overall cybersecurity to be evaluated. System and software engineering analysis does not consider the real threats and the ways their designs and selected controls can be compromised. In many cases, these acquisition and development technology teams do not include resources that understand how technology can be attacked, and they do not sufficiently consult external resources with this knowledge to help support the design analysis. By the time the separate security team sees the design, development is typically already underway. Design weaknesses are left in place because they are discovered too late to be corrected without major cost and schedule impacts.

Challenges for Evaluating Trusted Dependencies

Security dependencies relate to components that some build for others to use or connect with (i.e., inherited risk). The following are known areas of concern for inherited/dependency risk:

- Each dependency represents a risk (e.g., reuse, open source, collaboration environments).
- Dependency and trust decisions should be based on realistic assessments of the dependency's inherent risks and opportunities.
- Dependencies are not static, so trust decisions must be re-evaluated regularly to identify changes that warrant reconsideration.
- Using shared components to build technology applications and infrastructure increases the dependency on and potential mismatch of others' decisions (i.e., supply chain risk).

Layer on layer of reused software is integrated into systems based on the way the software is built and fielded. Code libraries allow developers to quickly assemble functionality, reusing language constructs and functionality constructs for frequently applied program actions. Tools for automating many of the development and testing steps are assembled into integrated development platforms (IDPs) for ease of use.

It is cost effective to reuse components and code previously developed for other projects to save time in development. However, that material was not created to meet the specifications of the current system and could have vulnerabilities introduced by the original developer; in these cases, the new system will inherit these vulnerabilities. Reuse practices that minimize the impact of this risk must be established. Since the current developers may not be familiar with the code, addressing vulnerabilities may be more time consuming than rewriting. In many cases, reuse is not cost effective when total costs are considered.

The reuse challenge also applies to commercial-off-the-shelf products and other third-party code used to address needed system functionality. Shared infrastructures, such as cloud environments, fall into this category. In some cases, the responsibility for addressing cybersecurity risk falls to programs when they are incorporating reused components into the system; sustainment planning for the system must reflect these issues. Using patching to address known vulnerabilities must be scheduled in a timely manner to minimize attacker opportunities—during both development and operations. In other cases, the responsibility of handling the risk must be spelled out in contracts or service level agreements (SLAs) established with the provider of services. We see lack of clarity in who is responsible for addressing risks from these sources and the timeliness in which cybersecurity issues will be addressed. Programs consider cost savings for services without establishing the critical mechanisms needed to ensure that the trust relationships that need to be established will operate with sufficient cybersecurity.

A cybersecurity strategy should provide plans for implementing and monitoring cybersecurity risk for each type of reuse implemented by a program. (Our experience is that, in practice, there is little planned, and response is primarily reactive.) Each plan involves establishing a vendor relationship that must be managed. Recent increases in supply chain risk show that this is an area where programs need to improve greatly.[4] Further information about inherited risk is available in an SEI white paper on reducing the attack surface for a system (Woody & Ellison, 2020b).

Challenges for Anticipating and Responding to Attacks

A growing and diverse population of adversaries uses both simple and increasingly sophisticated capabilities to compromise the confidentiality, integrity, and/or availability of technology assets. Adversaries often use a combination of technology, processes, standards, methods, and practices to craft a successful attack (i.e., socio-technical responses). Attacks are designed to take advantage of the ways users normally use technology or to contrive exceptional situations where users' defenses are circumvented. Attackers' profiles, capabilities, and methods are continually evolving.

To be successful, an attacker must bring together three key pieces:

- a weakness in the design, coding, or implementation of the system that can be triggered to promote unexpected and unwanted execution capabilities that can compromise confidentiality, availability, and/or integrity of the system
- access to that weakness through connected trusted systems, compromised authentication and authorization capabilities, or other attack vectors
- tools that enable the attacker to exploit the weakness, bypass security controls, and gain unauthorized system capabilities to trigger unwanted system behaviors

---

[4] See the SEI webcast, *SolarWinds Hack: Fallout, Prevention, and Recovery* (Software Engineering Institute, 2021).

Much attention is focused on potential vulnerabilities in code, and extensive effort has been expended to improve and implement a wide range of code analysis capabilities to reduce the availability of weaknesses for attackers to exploit. However, design weaknesses introduced in architecture and design implementation represent well over half of the top 25 common weaknesses and enumerations (CWEs) (Common Weakness Enumeration, 2020).

These weaknesses, which are prevalent in today's systems and software engineering, result from insufficient security requirements and insufficient understanding of how that technology can be compromised. Knowledge about the ways that technology can be compromised is still rarely taught in college and university curriculums; therefore, it may not be a part of the technical knowledge of someone with a background in computer science or system engineering.

As reuse continues to expand (see the "Challenges for Evaluating Trusted Dependencies" section above), broad availability of code increases through open source and acquisition sources as well as theft. The same tools that can be used to remove weaknesses can also be used by attackers to identify unmitigated weaknesses to exploit. There are also tools for reverse engineering operational modules to recovery the source. Versions of these tools are free, so the capabilities available to an attacker to discover and exploit weaknesses are extensive.

With both the level of software and the pool of potential exploits expanding, it is critical to update code as soon as new exploits are identified. Too much software was not designed and implemented to be effectively maintained for cybersecurity. Costly and cumbersome certification processes have not kept up with the realities of attacker capabilities, and some system engineers may not understand the attack potential of the software they implement. As a result, the cybersecurity strategy is too limited to address the critical realities of system vulnerabilities.

Challenges for Coordinating Security Throughout the Lifecycle

Assuring the security and resiliency of a system's mission-critical functions requires the following activities:

- Plan for what might go wrong; develop requirements for misuse/abuse cases with corresponding system and software requirements for secure, resilient response and operation.
- Build security and resiliency into the system.
- Verify that expected security and resiliency characteristics were actually built into the system.
- Clearly establish authority and responsibility for coordination at an appropriate level in the organization to ensure effective participation and coverage.

The lack of attention on misuse and abuse cases is increasingly a major gap in system requirements. Some programs leverage reusable components and shared platform capabilities as they rush to define functionality and identify ways that cost and schedule can be reduced to deliver results as quickly as possible. However, in their rush, they often fail to sufficiently specify how the system should perform under attack and compromise. Even when threat modeling is well established, system requirements lack the sufficient integration of threat knowledge to drive planned capabilities. Reliance is placed on selected compliance controls, but these are not tightly coupled with system operational capabilities, leaving gaps where unexpected behaviors and attempts to bypass controls are supported. Completeness of system capability is assigned to the system engineering team to ensure the system meets its requirements. However, from what we have seen, no one is formally given the responsibility for developing a complete set of misuse and abuse cases to ensure all reasonable threats are addressed.

With the transition to iterative development and an increased adoption of DevSecOps, the development approach and environments now become a permanent part of the operational system. These must

be in place to support the creation and fielding of new functionality and updates for long periods of time. Hence, they must be treated as part of the operational environment and hardened sufficiently to ensure they do not become conduits for attack of the system and its components. Traditionally, development environments do not have this level of rigor applied to them since they were thought of as temporary. For many programs, this perception of transience still remains.

For every system, cybersecurity is not the only critical quality. Reliability, maintainability, usability, safety, anti-tamper, and other system attributes, as well as cost and schedule, are equally important. Program management must establish how cybersecurity experts will collaborate with those addressing these other important attributes to define how tradeoffs will be managed and identified. Choices that impact mission risk should not be made without careful thought and consideration. We see too many programs that fail to establish coordination mechanisms among the various groups, managing each attribute and determining tradeoffs considering only cost and schedule. As a result, mission-critical capabilities can be jeopardized.

Challenges for Measuring to Improve Cybersecurity

Cybersecurity attributes of critical products, processes, and resources must be identified, measured, and monitored throughout the lifecycle. The best means for identifying metrics is by using the Goal-Question-Indicator Measure (GQIM) method, which identifies and defines meaningful indicators that align management, engineering, and improvement with business goals.[5]

The cybersecurity strategy should describe how a program will measure improvement. Opportunities would include:

- Use risk assessments that consider both current and potential threats, vulnerabilities, and impacts to identify critical goals for attributes to be measured and monitored.
- Develop and consistently apply well-formed measurement definitions and procedures to establish the credibility of the measurement and analysis results.
- Include all elements of the socio-technical environment that touch engineering and acquisition activities (e.g., processes, procedures, products, and resources).
- Support measurement with robust engineering planning; define a security measurement plan that spans the lifecycle, and develop requirements for any needed instrumentation.

Every activity within the lifecycle as well as each process and tool can produce data to feed a measurement effort. The challenge is determining where improvements in current practice would be useful for mission success and identifying the information that can help determine if a change has been successful.

We see programs measuring lots of individual activities, but many programs are not defining how these low-level metrics are useful to management to determine if expected cybersecurity results will be achieved. As a start, the strategy must define a vision of successful cybersecurity for the program. If the team assembled to address cybersecurity is sufficiently diverse, there will be several visions of success. The plan must consider ways to measure against each of these visions to provide a broader view of the potential goal. To date, we have seen many metrics collected, but little planning for what

---

5 SEI training materials are available for GQIM. For more information, see *Goal-Driven Measurement (IGDM) SEMA Course* on the SEI website (https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=635664).

is needed. Common activities we have seen include using checklists for compliance and counting vulnerabilities; however, these activities alone will not provide a means for measuring cybersecurity improvement.

Summary

A cybersecurity strategy is a critical planning document for program security. It must define the actions a program plans to perform to address cybersecurity. The strategy must clearly establish who is responsible for its success and the range of personnel resources that should be used to support its success. Those who implement the plan are a key element to its success. They must be knowledgeable in cybersecurity and system and software engineering and sufficiently focused on the need to provide results. They must be able to navigate the program's acquisition and development lifecycle and keep the plan current with tradeoffs that are made across the lifecycle that expand potential risk. These plan implementers must be invested in ensuring that cybersecurity is addressed for the program; they should not just be brought in periodically to glance at materials and sit through a few meetings.

The cybersecurity strategy will not remain static. Recognizing when changes are needed is a critical aspect of monitoring change and improvement. Metrics can play a valuable role in this aspect.

The strategy must cover more than just compliance; it must address the system's attack surface and make every effort to limit risks from all forms of technology (e.g., hardware, software, and firmware) and all sources of technology (e.g., reuse, third-party components, and external services).

On the whole, programs continue to react to cybersecurity instead of building it into all aspects of the system and its lifecycle. A well-planned cybersecurity strategy can help bridge this gap to support improved mission success.

References

Alberts, C. J., Woody, C., & Dorofee, A. J. (2014). Introduction to the Security Engineering Risk Analysis (SERA) Framework (CMU/SEI-2014-TN-025). Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=427321

Common Weakness Enumeration. (2020, August 20). 2020 CWE Top 25 Most Dangerous Software Weaknesses. https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html

Department of Defense. (2015, November 10). Violence in the workplace. Memorandum: Outline and Guidance for Acquisition Programs' Cybersecurity Strategies. https://www.acqnotes.com/wp-content/uploads/2014/09/DoD-CIO-Cybersecurity-Strategy-Outline-and-Guidance-10-Nov-15-1.pdf

Mead, N. R., Shoemaker, D., & Woody, C. (2013). Principles and Measurement Models for Software Assurance. International Journal of Software Engineering, 4(1). DOI: 10.4018/jsse.2013010101

Software Engineering Institute. (2021, February 10). SolarWinds Hack: Fallout, Recovery, and Prevention. [Webcast] Software Engineering Institute. https://www.sei.cmu.edu/news-events/events/event.cfm?customel_datapageid_5541=308529

Woody, C. & Creel, R. (2020, November 15). What Is Cybersecurity Engineering and Why Do I Need It? [Webinar] Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=650076

Woody, C. & Ellison, R. (2020a). Building a Cybersecurity Strategy. Special Issue on Rigor and Inter-Disciplinary Communication of the Journal of Systemics, Cybernetics, and Informatics (JSCI), 18(1), 206–216. http://www.iiisci.org/journal/sci/Contents.asp?var=&next=ISS2001

Woody, C. & Ellison, R. (2020b). Attack Surface Analysis - Reduce System and Organizational Risk. Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=650193

Author Biographies

Dr. Carol Woody, a principal researcher for the CERT Division of the Software Engineering Institute at Carnegie Mellon University, is building capabilities and competencies for measuring, managing, and sustaining cybersecurity for highly complex software intensive systems. She has successfully implemented solutions in many domains, including banking, mining, manufacturing, government, and finance. She co-authored the book *Cyber Security Engineering: A Practical Approach for Systems and Software Assurance*, published by Pearson Education as part of the SEI Series in Software Engineering. The CERT Cybersecurity Engineering and Software Assurance Professional Certificate, released in March 2018, is based on the research she led.

Rita Creel is acting deputy director for the CERT Division of the Software Engineering Institute at Carnegie Mellon University. She has over 25 years of experience in software-intensive systems engineering and acquisition, cybersecurity, and systems and software measurement and analysis.

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.