Carnegie Mellon University
Software Engineering Institute

AN APPROACH FOR INTEGRATING THE SECURITY ENGINEERING RISK ANALYSIS (SERA) METHOD WITH THREAT MODELING

Christopher Alberts and Carol Woody

January 2019

Introduction

In today's operational environments, multiple organizations often are required to work collaboratively in pursuit of a single mission, creating management complexity that is difficult to manage effectively. Successful execution of a multi-organizational mission demands management approaches that effectively coordinate task execution and risk management activities among all participating groups. Organizations across government and industry are beginning to implement mission assurance programs in an effort to coordinate mission execution and help ensure mission success.

The Department of Defense (DoD) Directive 3020.40, titled *Mission Assurance (MA)*, defines *mission assurance* as "a process to protect or ensure the continued function and resilience of capabilities and assets, including personnel, equipment, facilities, networks, information and information systems, infrastructure, and supply chains, critical to the execution of DoD mission-essential functions in any operating environment or condition" [DoD 2018]. This directive requires DoD Components to prioritize mission assurance efforts in support of critical DoD strategic missions. Weapon system acquisition falls outside the direct scope of the DoD's mission assurance directive. However, the directive does state important concerns relevant to all systems that

- risk management must be addressed as early as possible in the acquisition of information technology across the lifecycle
- acquisition programs must integrate mission assurance goals and activities with acquisition guidance

As a result, mission assurance must be considered during the acquisition of DoD software-intensive systems, such as weapon systems. From a cyber perspective, acquisition programs should start managing cybersecurity risk early in the system acquisition lifecycle. A complicating factor is that most software-intensive systems are networked. While networking offers many operational efficiencies to a system's stakeholders, it also expands a system's cyber-risk profile.

A network of independently managed software-intensive systems, referred to as a system of systems (SoS), provides information and services that are essential for successful mission execution. Cyber attacks with the potential for mission impact can target any system within an SoS environment, creating complex attack vectors that must be considered during cyber-risk analysis. Cyber attacks are designed to exploit weaknesses and vulnerabilities in a system's software components, which makes software the focal point for early lifecycle cyber-risk analysis. Software must be designed and architected with the

knowledge that it must function as intended in an increasingly contested, challenging, and interconnected cyber environment. Software assurance is thus essential for achieving mission assurance.

Software assurance is defined as a level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, throughout the lifecycle [NIA 2010]. Software assurance is becoming increasingly important to organizations across all sectors because of software's increasing influence in business- and mission-critical systems. For example, consider how the size of flight software[1] has increased over the years. Between 1960 and 2000, the degree of functionality provided by software to the pilots of military aircraft has increased from 8% to 80%. At the same time, the size of software in military aircraft has grown from 1,000 lines of code in the F-4A to 1.7 million lines of code in the F-22. This growth trend is expected to continue over time [NASA 2009]. As software exerts more control of complex systems, like military aircraft, the potential risk posed by cybersecurity vulnerabilities will increase in kind.

In 2005, researchers from the CERT® Division at Carnegie Mellon® University's Software Engineering Institute (SEI) started investigating how to enable mission assurance in SoS environments [Alberts 2005]. A major conclusion of this research was that system-oriented cyber-risk methods do not readily scale to SoS environments. New analysis approaches were needed to complement traditional system-oriented analysis activities.

In 2013, CERT researchers started investigating how to conduct cyber-risk analysis early in the acquisition lifecycle (i.e., during requirements, architecture, and design). The product of this research project was the Security Engineering Risk Analysis (SERA) Method, a scenario-based approach for analyzing complex cybersecurity risks early in the acquisition lifecycle. The SERA Method integrates system and software engineering with operational security by requiring engineers to analyze operational cybersecurity risks as software-reliant systems are acquired and engineered. The method analyzes data provided by a program's existing system analysis activities and translates that data into a mission context. Our overarching goal when developing the SERA Method was to define a risk-based approach for evaluating mission assurance in SoS environments. This mission focus differentiates the SERA Method from other traditional system-focused cybersecurity methods.

In this report, we explore the link between the SERA Method and threat modeling, which has become a popular engineering practice across industry and government organizations in the past decade. A threat modeling method defines an approach for identifying countermeasures that can be engineered into a software system. In this report, we specifically examine how the system-focused cybersecurity data generated by a threat modeling method can be integrated into a mission assurance context using the SERA Method. We begin by defining the broad problem space addressed by the SERA Method.

---

[1] Flight software is a type of embedded real-time software used in avionics.

® CERT and Carnegie Mellon are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Problem Space

Our initial research-and-development goal for the SERA Method was to develop an approach capable of analyzing the complexity of modern cybersecurity risks. Sources of cybersecurity risk originate in the complex network of people, processes, and technologies that form the foundation of today's operational and mission environments. We use the following perspectives to describe the nature of these environments:

- mission
- system of systems
- socio-technical
- cyber-physical
- software

These perspectives are important because they influence how cybersecurity risk analysis must be performed. As a result, the five perspectives define the problem space for our research-and-development activities. Each perspective is described in the remainder of this section, beginning with the mission perspective.

Mission Perspective: A *mission* is a fundamental objective or purpose being pursued by an individual, group, or organization. People, processes, and technologies are then organized in a manner to achieve the mission. This perspective highlights the effect of cybersecurity attacks on the mission that an individual, group, or organization is pursuing. As a result, a cyber-risk analysis must extend beyond the boundary of a single technical system and consider the impact on the mission.

System-of-Systems Perspective: A *system of systems (SoS)* is defined as a set or arrangement of interdependent systems that are related or connected (i.e., networked) to provide a given capability [Levine 2003]. A key characteristic of an SoS is managerial independence, where each system within an SoS is managed independently from the other systems [Maier 1996]. The SoS perspective describes how a software-reliant system must function as part of a multi-system environment to achieve stakeholders' objectives. This complex, multi-system environment has implications for how cybersecurity must be analyzed and managed. The SoS perspective also illustrates the complex nature of cybersecurity attacks and how they typically include many systems that are managed by multiple, independent organizational entities.

Socio-Technical Perspective: A *socio-technical system* is defined as interrelated technical and social elements that are engaged in goal-oriented behavior. Elements of a socio-technical system include the people who are organized in teams or departments to do their work tasks and the technologies on which people rely when performing work tasks. This perspective stresses the prominent role of people in creating, using, and maintaining technologies. It also highlights the role of people in causing and preventing cybersecurity attacks.

Cyber-Physical Perspective: A *cyber-physical system* is an engineered system that is built from, and depends upon, the seamless integration of computational algorithms and physical components. Cyber-physical systems merge the physical and virtual worlds, integrating objects, data, and services. Cyber

processes monitor and collect data from physical processes, such as the steering of an automobile or the observation of vital signs of a hospital patient. Cyber-physical systems are networked, making their data globally available to other processes. These systems thus make it possible for software to directly interact with events in the physical world. The cyber-physical perspective emphasizes the notion that cybersecurity attacks can produce consequences in the physical world.

Software Perspective: A *software-reliant system* is a system whose behavior (e.g., functionality, performance, safety, security, interoperability) is dependent on software in some significant way [Bergey 2009]. The software perspective is focused on building security controls into software, not treating security as an add-on feature that will be addressed during software sustainment activities. This perspective requires addressing security concerns from the earliest phases of the system and software lifecycles through the sustainment and evolution of deployed software-reliant systems.

Traditional cyber-risk analysis methods generally incorporate one or two of the above perspectives. In contrast, we designed the SERA Method to address all five perspectives.

SERA Method Overview

The SERA Method defines a scenario-based approach for analyzing complex cybersecurity risks in software-reliant systems and systems of systems across the lifecycle and supply chain [Alberts 2016]. 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 then use those scenarios to focus its limited resources on controlling the most significant security risks.

The SERA Method can be self-applied by the person or group that is responsible for acquiring and developing a software-reliant system or facilitated by external parties on behalf of the responsible person or group. In either case, a small team of approximately three to five people, called the *Analysis Team*, is responsible for implementing the method and reporting findings to stakeholders.

An Analysis Team is an interdisciplinary team that requires members with diverse skill sets. Examples of skills and experience that should be considered when forming a team include: security-engineering risk analysis, systems engineering, software engineering, operational cybersecurity and physical/facility security. The exact composition of an Analysis Team depends on the point in the lifecycle in which the SERA Method is being applied and the nature of the engineering activity being pursued. Table 1 highlights the four tasks that the Analysis Team performs when conducting the method [Alberts 2016].

Table 1: SERA Method Overview

Task	Description	Outputs
1. Establish operational context	Task 1 defines the operational context for the analysis. The Analysis Team compiles/develops operational views that (1) define the mission and (2) document how systems and software support the execution of that mission. Task 1 sets the context and scope for the subsequent risk analysis activities. Task 1 is important because it defines a performance baseline for each selected mission. This baseline establishes what is considered to be normal operational performance during mission execution as well as the systems and software components that support the mission execution. The Analysis Team determines which systems are of highest priority for further examination. These high-priority systems are called systems of interest. Cyber risks affecting each system of interest are subsequently analyzed in relation to the baseline of operational performance.	• mission thread(s) • SoS diagram(s) • system architecture diagrams • software architecture diagrams • dataflow diagrams • use cases • data security attributes • network topology diagrams • other diagrams as appropriate
2. Identify risk	Task 2 defines the cyber-risk identification activities for the SERA Method. Here, the Analysis Team transforms security concerns into distinct, tangible risk scenarios that can be described and measured. The Analysis Team reviews the operational context documented in Task 1 as well as relevant system-specific data for each system of interest. The team then develops a set of cyber-risk scenarios for each selected system of interest.	• cyber-risk scenarios
3. Analyze risk	Task 3 focuses on the cyber-risk analysis. The Analysis Team evaluates each cyber-risk scenario in relation to predefined criteria to determine its probability, impact, and risk exposure.	• risk measures (probability, impact, risk exposure) for cyber-risk scenarios
4. Develop control plan	Task 4 establishes a plan for controlling a selected set of risks. The Analysis Team prioritizes the cyber-risk scenarios based on their risk measures. The team then determines the basic approach for controlling each risk (i.e., accept or plan) based on predefined criteria and current constraints (e.g., resources and funding available for control activities). Finally, the Analysis Team develops a control plan for each risk that is not accepted.	• prioritized cyber-risk scenarios • control plan for each high-priority cyber-risk scenario

Two features of the SERA Method differentiate it from traditional cyber-risk analysis methods: (1) documenting an explicit line of sight from mission to software and (2) implementing a scenario-based analysis approach. The next two sections explore these key differentiators, beginning with the line of sight from mission to software.

SERA Method: A Line of Sight from Mission to Software

The SERA Method incorporates a top-down analysis approach. As illustrated in Figure 1, the goal of the SERA Method's initial task is to establish a line of sight from a mission to the software components that support it. The starting point of this top-down approach is the selection of mission threads that will be included in the analysis. A *mission thread* is a sequence of end-to-end activities and events that takes place to accomplish a specific result. A mission thread documents the actors, tasks, inputs, and outputs required to achieve stakeholders' objectives. Actors in a mission thread can be people, groups of people, processes, or technologies. Mission threads often include a mixture of different types of actors. For example, people can perform some activities in a given mission thread, while technologies perform others. Diagramming techniques, such as workflow diagrams or process maps, are often used to document mission threads. The mission perspective, as depicted by a mission thread, is important because it defines the operational context of a cyber-risk analysis.

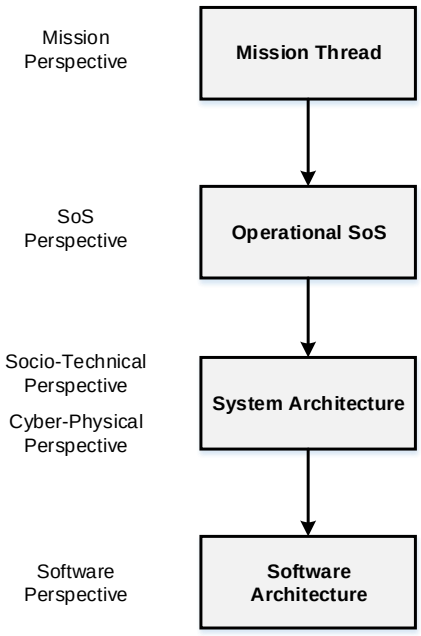

Figure 1: SERA Task 1—Documenting a Line of Sight from Mission to Software

Based on the mission thread, analysts can identify which systems are needed to execute the mission thread. This collection of systems that supports mission execution is called the *operational SoS*. Some of these systems may serve as actors in the mission thread and be responsible for performing mission thread activities. Other systems in the operational SoS might support mission thread activities performed by people. As a result, systems within the operational SoS directly or indirectly support the execution of mission thread activities, providing an SoS perspective that is critical for achieving mission success. Various diagramming and modeling techniques can be used to represent an operational SoS, including network topologies, inter-system communication diagrams, and inter-system dataflow diagrams.

Once the operational SoS is defined, analysts need to make additional scoping decisions. It is not practical to analyze all systems in detail. Analysts must determine which systems are of highest priority for further examination. The selection of which systems to analyze in greater detail generally occurs in one of two ways. First, analysts can identify high-value systems in the SoS that might be the target of a cyber attack (i.e., identify critical assets). The subsequent cyber-risk analysis focuses on those high-value systems. Each high-value, or critical, system is referred to as a *system of interest*. Alternatively, analysts might be asked to analyze a specific system within the SoS. For example, a program manager that is acquiring a replacement for a legacy system might want to conduct a cyber-risk analysis of the technology being acquired. In this case, the acquired system is referred to as the *system of interest* and becomes the focus of the subsequent cyber-risk analysis. Additional information must be collected or compiled for each system of interest that is selected for analysis, beginning with information about the system architecture.

A *system architecture* is a conceptual model that defines the structure and behavior of a system. It provides a representation of a system that maps system functionality to hardware and software components. It also shows how people interact with hardware and software components and defines key interfaces for the system. System architecture documentation generally includes multiple views of a system, such as hardware diagrams, intra-system networking and communication diagrams, intra-system dataflows, use cases, and system interface diagrams. The multiple views of a system architecture help to describe the socio-technical and cyber-physical aspects of a software-reliant system.

Finally, a *software architecture* is a conceptual model that defines the detailed structure of a system's software components, the behaviors of those components, and the relationships among the components. Software architecture documentation generally includes multiple views of a system, such as dataflow diagrams, use cases, and software interface diagrams. Cyber attacks exploit weaknesses and vulnerabilities in a system's software components. As a result, the software perspective provides the starting point for cyber-risk identification.

SERA Method: SoS Cyber-Risk Scenarios

Task 2 of the SERA Method is focused on cyber-risk identification. The SERA Method uses scenarios to describe cybersecurity risks. Our early piloting of the SERA Method showed that scenarios are effective for capturing the complexities and nuances of a cybersecurity risks in SoS environments [Alberts 2016]. A *cyber-risk scenario* tells a story of how one or more threat actors can cause adverse mission consequences for stakeholders by exploiting vulnerabilities in one or more software-reliant systems. A SERA cyber-risk scenario comprises the following three elements:

1. *SoS cyber-attack vector*—how that a threat actor traverses an SoS environment to gain access to the target of the cyber attack (e.g., software component). An SoS cyber-attack vector is also referred to as an *access path*.

2. *cyber attack*—steps that a threat actor takes to launch a cyber attack on the selected target and a description of the direct outcome of the attack (e.g., data disclosure, data modification, data unavailability)
3. *consequences*—description of a cyber attack's impact on the mission (based on mission thread analysis)

Analysts create cyber-risk scenarios by reviewing the operational context documented in Task 1of the SERA Method as well as relevant system-specific data for each system of interest. System-specific data typically includes the results of safety assessments, security assessments, and risk assessments that have been documented for each system of interest. Figure 2 shows how the three elements of cyber-risk scenario map to operational context defined in Task 1 of the SERA Method.

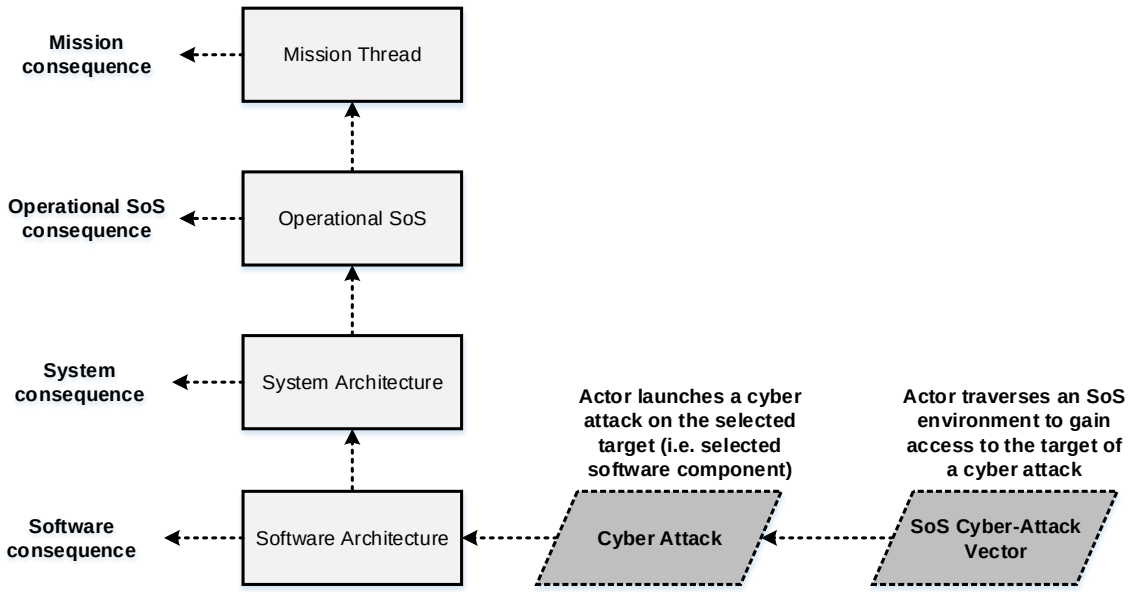

Figure 2: SERA Task 2—Cyber-Risk Scenarios

The end goal of the malicious actor in Figure 2 is to disrupt the mission thread. From the actor's perspective, a successful cyber attack will lead to mission failure. To produce the desired mission consequence via a cyber attack, the actor must first gain access to a high-value software component that has a line of sight to the mission thread. Malicious actors often must traverse circuitous routes though an SoS environment to gain access to high-value software components. Many of the systems in an SoS cyber-attack vector do not directly or indirectly link to the mission thread. In other words, most software components that are exploited in a given attack vector do not have an obvious line of sight to the targeted mission thread. These software components might be part of enterprise information systems, supply chain systems, development systems, and test systems, among others. Network topology diagrams are useful for identifying SoS cyber-attack vectors.

Once an actor gains access to a high-value software component, he or she is in position to execute a cyber attack on that component. The immediate goal of the cyber attack is to affect data that are stored,

processed, or transmitted by the high-value software component. Cyber attacks generally produce one of the following direct consequences affecting the confidentiality, integrity, or availability of data:

- data disclosure (confidentiality)
- data modification (integrity)
- insertion of false data (integrity)
- destruction of data (availability)
- interruption of access to data (availability)

Figure 2 uses the term *software consequence* when referring to the direct consequence of an attack. Understanding the direct consequence of a cyber attack is important. However, risk is ultimately assessed using the indirect consequences of an attack. Therefore, the impacts on the system, operational SoS, and mission thread must be determined. The data documented in Task 1 of the SERA Method provides the context for risk identification in Task 2, where analysts develop a set of cyber-risk scenarios for each selected system of interest. Task 3 builds on the results of Task 2 by requiring analysts to evaluate the probability, impact, and risk exposure of each cyber-risk scenario. Finally, during Task 4, analysts establish a plan for controlling each high-priority cyber-risk scenario, marking the conclusion of the SERA Method.

To this point in the report, we have focused exclusively on the SERA Method. The next section shifts the focus to threat modeling, where we provide a brief overview of a few common threat modeling methods and examine the scope typically addressed by threat modeling methods.

Threat Modeling Methods

A threat modeling method is an approach for creating an abstraction of a software system, profiling an attacker's abilities and goals, and identifying possible threats that the system must mitigate [Mead 2018, Shevchenko 2018]. Threat modeling methods are used by a variety of organizations across industry and government. In the past decade, several threat modeling methods have been developed. Some methods are based on National Institute of Standards and Technology (NIST) standards, such as the Risk Management Framework (RMF), while other employ checklists [Mead 2018]. Examples of some common threat modeling methods include [Mead 2018, Shevchenko 2018]:

- *STRIDE*[2]—developed and implemented at Microsoft. STRIDE models a system and its related dataflows.

[2] STRIDE is an acronym for the types of threats considered by the method: spoofing identify, tampering, repudiation, information disclosure, denial of service, and elevation of privilege.

- *Trike*—developed to improve on perceived deficiencies of STRIDE. Trike is designed for security auditing, and it models threats from a defensive perspective (as opposed to an attacker's perspective).
- *Process for Attack Simulation and Threat Analysis (PASTA)*—implements a seven-stage process that determines the impact of threats to an application and business. PASTA is designed to consider both business and technical requirements.
- *Visual, Agile, and Simple Threat Modeling (VAST)*—based on an automated threat modeling platform and intended for large/medium organizations. VAST produces application threat models and operational threat models, which enables the integration of VAST into an organization's Agile or DevOps lifecycle.
- *Security Cards*—relies on physical cards to assist brainstorming about potential cyber threats. The Security Cards method facilitates the exploration of potential security threats for a particular system and helps to develop a security mindset within an organization.
- *Misuse Cases*—extends use case diagrams to include security threats and countermeasures. Misuse cases depict the interactions between malicious actors and vulnerabilities within the context of a use case.
- *Personae non Grata (PnG)*— characterizes users as archetypes that exhibit undesired and malicious behaviors. PnG forces analysts to view the system from an attacker's point of view by focusing on how archetypal users can misuse a system.

The above list is not compressive; it lists a few methods to illustrate the diversity of approaches being implemented by acquisition and engineering organizations. Readers should consult other publications (e.g., [Mead 2018] and [Shevchenko 2018]) to get a more comprehensive survey of available threat modeling methods. Despite the range of options, most threat modeling methods are focused on analyzing cyber-threats to software-intensive systems. This concept is illustrated in Figure 3.

As depicted in Figure 3, threat modeling methods are system-oriented. They consider a subset of the information contained in the SERA Method's cyber-risk scenarios (as depicted in Figure 2). For example, threat modeling methods typically do not analyze SoS cyber-attack vectors, the impact on the operational SoS, and the impact on the associated mission thread, all of which are included in SERA cyber-risk scenarios.

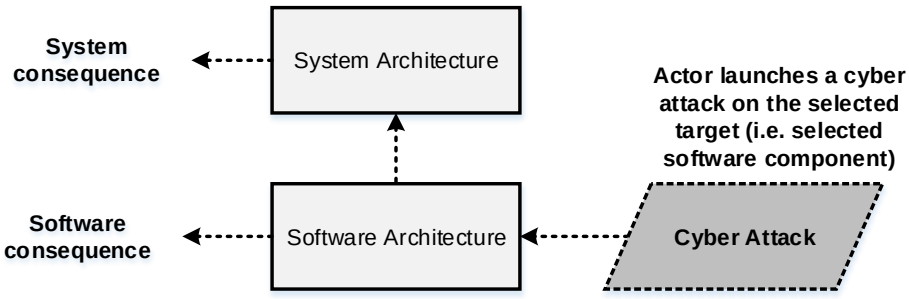

Figure 3: Focus of Threat Modeling Methods

The SERA Method is designed to analyze cybersecurity risk at multiple level of abstraction. The method provides the capability to analyze cybersecurity risk in a software-intensive system, operational SoS, and mission thread. As a result, the SERA Method includes a threat modeling component that provides the functionality depicted in Figure 3. However, many organizations have already adopted threat modeling methods, such as those methods on the above list. Alternatively, we have encountered organizations that developed their own unique versions of threat modeling (often combining characteristics of multiple methods listed above) to better meet the needs of organizational stakeholders. We designed the SERA Method to leverage existing system assessment data, including the results of an organization's preferred threat modeling method. This topic is explored in the next section.

Integrating SERA with Threat Modeling

Acquisition programs perform many different types of cybersecurity assessments, ranging from running vulnerability scanning tools on software code to assessing how well NIST RMF controls have been implemented in the system being acquired. When we apply the SERA Method for an acquisition program, we review the results of all relevant system-oriented assessments conducted by program and contractor personnel and translate that data into a mission context. The following assessments typically provide data that are used as input to the SERA Method:

- threat modeling
- NIST RMF assessments
- cyber risk assessments
- threat assessments
- vulnerability assessments
- safety assessments

One of the most important sources of system assessment data is threat modeling. Figure 4 shows how the SERA Method *integrate*s data generated by a threat modeling method into the broader mission assurance context. As illustrated in the figure, a threat modeling method provides data about potential cyber attacks to a system of interest as well as the direct consequences to the system of interest. That data must be reviewed, filtered, processed, and formatted to provide seamless integration into the SERA Method. Our field experience has shown that it is often necessary to fill gaps in a threat modeling dataset to address incorrect, inconsistent, or incomplete data, and the SERA Method provides a means of filling those gaps. Overall, the approach depicted in Figure 4 has worked well in practice.

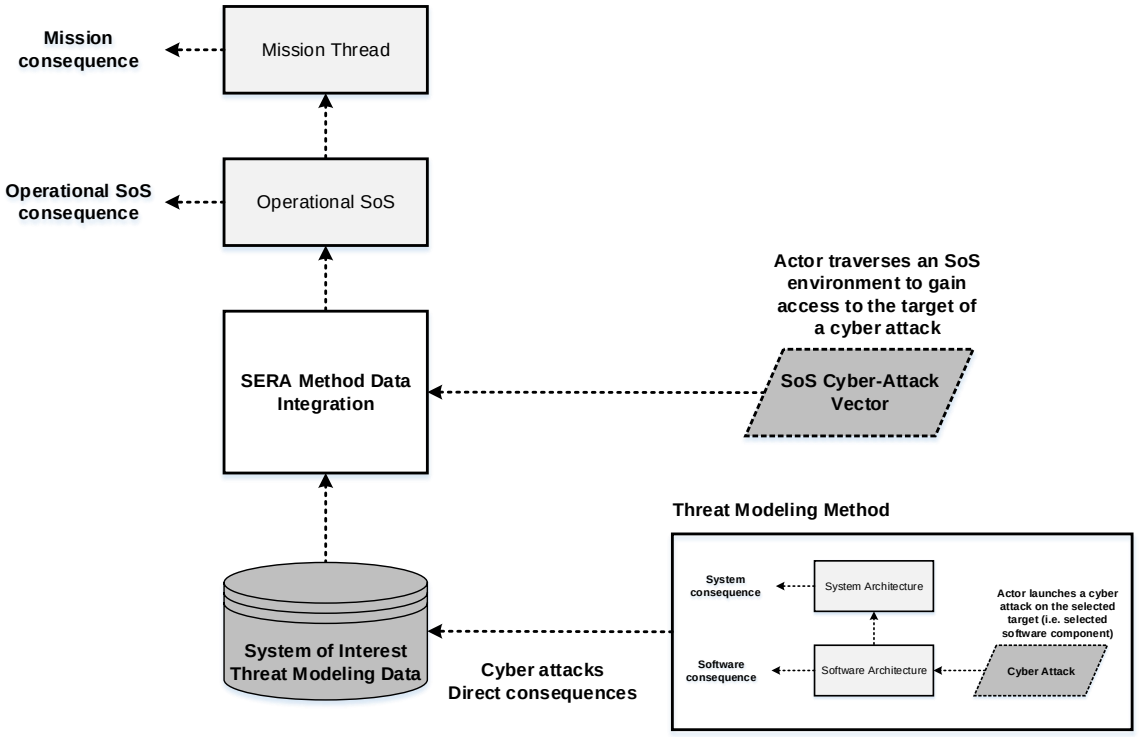

Figure 4: Integrating Threat Modeling Data into the SERA Method

The core objective of the SERA Method is to evaluate mission assurance in SoS environments. Previous SEI research into mission assurance showed that traditional, system-oriented risk management approaches are unable to handle the complex risks inherent in SoS environments [Alberts 2009]. System-oriented cyber-risk methods, such as threat modeling, are effective at assessing how cyber attacks can adversely affect a given system's performance. However, these methods (1) cannot identify risks derived from the interactions of multiple systems within an SoS environment and (2) lack sufficient context to estimate mission consequences. The SERA Method complements system-oriented methods by providing a means of translating system data into a mission assurance context.

Summary

Mission assurance provides a level of confidence in the potential for achieving mission success. The complexity inherent in SoS environments tends to increase the overall cyber risk to a mission, and a high degree of risk corresponds to low assurance. Effective cyber-risk management across the acquisition lifecycle is essential for establishing and maintaining mission assurance over time.

SEI researchers developed the SERA method to analyze complex cybersecurity risks early in the acquisition lifecycle. The method is designed to integrate data provided by a program's existing system analysis activities, including threat modeling, and translate that data into a mission assurance context.

We have conducted multiple SERA pilots for DoD and government acquisition programs over the past several years, looking for gaps in system requirements and weaknesses in a system's architecture and design. In all pilots, we identified cyber risks that were missed by system-focused cybersecurity methods. In some pilots, the SERA results identified gaps in a system's security requirements (e.g., missing, incomplete, or incorrect requirements). In other pilots, we identified weaknesses in the software/system architecture or design that could lead to adverse mission consequences. In general, our field experience in applying the SERA Method indicates that the method's focus on mission assurance in SoS environments is useful for reducing residual cyber risk in deployed software-intensive systems.

For this report, we examined several threat modeling methods to determine how they could be used in conjunction with the SERA Method. We concluded that the SERA Method can integrate the system-focused threat data generated by threat modeling methods. When translating threat modeling data into the mission context, the SERA Method enables analysts to address any incorrect, inconsistent, or incomplete data provided by the threat modeling method. Finally, if an acquisition program has not implemented a preferred threat modeling practice, the program's engineering team can use the SERA Method's threat modeling component to generate the required system threat data. Overall, we conclude that the SERA Method is beneficial for expanding the narrowly focused scope of threat modeling into a mission assurance context.

References

[Alberts 2005]

Alberts, Christopher & Dorofee, Audrey. *Mission Assurance Analysis Protocol (MAAP): Assessing Risk in Complex Environments* (CMU/SEI-2005-TN-032). Software Engineering Institute, Carnegie Mellon University, 2005. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=7505

[Alberts 2009]

Alberts, Christopher & Dorofee, Audrey. *A Framework for Categorizing Key Drivers of Risk* (CMU/SEI-2009-TR-007). Software Engineering Institute, Carnegie Mellon University, 2009. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9093

[Alberts 2014]

Alberts, Christopher; Woody, Carol; & Dorofee, Audrey. *Introduction to the Security Engineering Risk Analysis (SERA) Framework* (CMU/SEI-2014-TN-025). Software Engineering Institute, Carnegie Mellon University, 2014. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=427321

[Alberts 2016]

Alberts, Christopher; Dorofee, Audrey & Woody, Carol. *Wireless Emergency Alerts Commercial Mobile Service Provider (CMSP) Cybersecurity Guidelines* (CMU/SEI-2016-SR-009). Software Engineering Institute, Carnegie Mellon University, 2016. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=463988

[Bergey 2009]
Bergey, John K. *A Proactive Means for Incorporating a Software Architecture Evaluation in a DoD System Acquisition* (CMU/SEI-2009-TN-004). Software Engineering Institute, Carnegie Mellon University, 2009. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=8935

[DoD 2018]
Office of the Secretary of Defense for Policy. *Mission Assurance (MA)* (DoD Directive 3020.40). Washington, DC, 2018. https://fas.org/irp/doddir/dod/d3020_40.pdf

[Levine 2003]
Levine, Linda; Meyers, B. Craig; Morris, Ed; Place, Patrick R. H.; & Plakosh, Daniel. *Proceedings of the System of Systems Interoperability Workshop (February 2003)* (CMU/SEI-2003-TN-016). Software Engineering Institute, Carnegie Mellon University, 2003. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=6469

[Maier 1996]
Maier, Mark. "Architecting Principles for Systems-of-Systems." 567-574. *Proceedings of the Sixth Annual International Symposium of INCOSE*. Boston, MA, July 7-11, 1996. INCOSE, 1996.

[Mead 2018]
Mead, Nancy R.; Shull, Forrest; Vemuru, Krishnamurthy; & Villadsen, Ole. *A Hybrid Threat Modeling Method* (CMU/SEI-2018-TN-002). Software Engineering Institute, Carnegie Mellon University, 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=516617

[NASA 2009]
National Aeronautics and Space Administration (NASA). *Final Report, NASA Study on Flight Software Complexity*. NASA Jet Propulsion Laboratory, Systems and Software Division, Pasadena, CA, 2009. http://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf

[NIA 2010]
Committee on National Security Systems. *National Information Assurance (IA) Glossary CNSS Instruction* (CNSS Instruction No. 4009). Fort George G. Meade, MD, 2010. http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

[Shevchenko 2018]
Shevchenko, Nataliya; Chick, Timothy A.; O'Riordan, Paige; Scanlon, Thomas Patrick; & Woody, Carol. *Threat Modeling: A Summary of Available Methods*. Software Engineering Institute, Carnegie Mellon University, 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=524448

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.

DM19-0056