Security Engineering Risk Analysis (SERA)

Carol Woody
CERT Division
Software Engineering Institute (SEI)
Carnegie Mellon University
+1-412-770-5133
cwoody@cert.org

ABSTRACT

In this presentation, I describe how the SEI's Security Engineering Risk Analysis (SERA) method provides a structure that connects desired system functionality with the underlying software to evaluate the sufficiency of requirements for software security and the potential operational security risks based on mission impact.

CCS Concepts

[Software and its engineering](#) → Software creation and management

[Software and its engineering](#) → Software organization and properties

[Security and privacy](#) → Software and application security

[Security and privacy](#) → Systems security

Keywords

Security risk analysis; software security engineering; security risk management; security lifecycle; supply chain risk management; vulnerabilities;

1. TOPIC DESCRIPTION

Researchers from Software Engineering Institute (SEI) developed the Security Engineering Risk Analysis (SERA) Method and successfully used SERA for analysis of operational security risks early in the lifecycle to identify design weaknesses and establish confidence for system and software assurance. The SERA Method is a model-based approach for analyzing complex security risks in software-reliant systems and systems of systems across the lifecycle and supply chain. Addressing security risk in this context requires consideration of the role of each system in addressing an operational mission and the impact of security issues on the specific system of interest that may originate in other mission-dependent systems.

2. VALUE OF TOPIC

This research focuses on the convergence of three major areas—systems engineering, software security engineering, and security risk management—and the value in using a cross-discipline approach for security. From a realistic mission thread, multiple system and software relationships and dependencies as well as critical assets and high-impact points of failure can be identified and mitigation approaches developed and validated.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author. Copyright is held by the owner/author(s).

SER&IP'16, May 17-17 2016, Austin, TX, USA

ACM 978-1-4503-4170-7/16/05.

<http://dx.doi.org/10.1145/2897022.2897024>

Operational security vulnerabilities generally have three main causes: (1) design weaknesses, (2) implementation/coding vulnerabilities, and (3) system configuration errors. Addressing design weaknesses as soon as possible is especially important because these weaknesses are not corrected easily after a system has been deployed. Remediation normally requires extensive redesign of the system, which is costly and often proves to be impractical.

Determining the importance of a security risk requires connecting it to mission impact. The SERA method provides systems engineers with a structure to connect desired system functionality with the underlying software to evaluate the sufficiency of requirements for software security and the potential operational security risks based on mission impact.

3. PRESENTATION DESCRIPTION

The SEI successfully applied this approach for alert originators as they planned for use of the Wireless Emergency Alerts (WEA) system, a public alerting service offered by the Department of Homeland Security starting in the spring of 2014. Mission threads provided a means for identifying and evaluating potential security threats in a way that non-technical stakeholders could understand the risks and determine appropriate responses.

A mission thread is an end-to-end set of steps that illustrate the technology and people resources needed to deliver expected behavior under a set of conditions and provide a basis for identifying and analyzing potential problems that could represent risks. For each mission step, the expected actions, outcomes, and assets are assembled. Confirmation that the components appropriately respond to expected operational use increases confidence that the system will function as intended even in the event of an attack [1].

An example WEA mission thread is shown in Figure 1. Each bar in the diagram represents an independent organization providing technology that can address the desired mission only when linked with the technology provided by the other participants in this pipeline. No one controls the mission thread and yet decisions made by each participant affect mission results and can impact all of the other participants.

Figure 1. WEA Mission Thread Diagram

Using a threat modeling technique such as STRIDE, which was developed by Microsoft [2], system engineers can analyze each step to determine the assets that need protection and ways in which they can be threatened.

The following table provides an example for step 4 when the operator establishes access to the alert origination system in the mission thread shown in Figure 1.

Table 1. Example for Step Four in Figure 1

Step Description	Assets	STRIDE Threat Identification Examples
Operator attempts to log on to the alert origination system.	<ul style="list-style-type: none"> One person Server (valid accounts/authentication information) Logon procedure Logon application Username/password data in database Communications between logon software, server, and alert origination. 	<ul style="list-style-type: none"> Unidentified individual attempts to log on with AOS operator's information Operator denies having logged on Capture of logon information using key logger or packet sniffer Operator's account not registered or servers are down Successful logon by an unidentified and unauthorized individual

From this information, security experts can develop realistic attack scenarios that characterize mission risks each participating system needs to be prepared to handle, in a form meaningful to stakeholders and mission-thread participants.

Figure 2 provides an example where an attacker steals a certificate of authentication in Step 4 and uses it to later send valid but illegitimate WEA messages, sending people into a harmful situation

Figure 2. WEA Mission Thread Risk.

Based on the success of this approach in helping alert originators understand the importance of addressing security risks, DHS supported a second pilot for SERA to evaluate another portion of the WEA mission thread—commercial mobile service providers. For this analysis, the existing infrastructure was evaluated to determine if appropriate security controls were in place to address both security risks within the direct control of CMSPs and those that might impact them based on actions taken by alert originators.

This presentation provides an overview of the SERA methodology [3], shares highlights of the two applications of SERA, the process developed to conduct the research, and the ways in which this research provided models to enable a cross-discipline approach.

4. ACKNOWLEDGEMENT

This material is based upon work funded and supported by Department of Homeland Security under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Department of Homeland Security or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0003380

5. REFERENCES

- [1] R. Ellison, J. Goodenough, C. Weinstock, and C. Woody. "Survivability assurance for system of systems" (CMU/SEI-2008-TR-008). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2008. <http://www.sei.cmu.edu/library/abstracts/reports/08tr008.cfm>
- [2] Microsoft Develop Network (MSDN). "The Strike Threat Model." Redmond, WA: MSDN, 2005. <http://msdn.microsoft.com/en-US/library/ee823878%28v=cs.20%29.aspx>.
- [3] Alberts. Christopher, Woody. Carol, and Dorofee. Audrey, "Introduction to the Security Engineering Risk Analysis (SERA) Framework," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Note CMU/SEI-2014-TN-025, 2014. <http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=427321>.