Volume 5, No. 2, March 2014 (Special Issue)
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Risk Based Analysis for Software Security

T.Gnana Krishna Deep[1], V.Venkata Koundinya[2], K.V.D.Kiran[3], Dr.L.S.S.Reddy[4]
2/4 B.Tech CSE[1], [2]
Koneru Lakshmaiah Education Foundation- K L University
Green Fields, Vaddeswaram, Guntur dist.
Andhra Pradesh, India-522502
deep.tipirneni@gmail.com[1], koundi12101994@gmail.com[2], kiran_cse@kluniversity.in[3]

***Abstract*--**For some years, the software engineering community has been working to identify practices aimed at emerging more secure software. Although some initial work has been done, efforts to measure software security statement have yet to develop in any substantive fashion. As a result development program and project managers lack poise in the security features of their software-reliant systems. This project is to advance the state-of-the-practice in software security measurement and analysis. This project is exploring how to use risk analysis to direct a society's software security measurement and analysis efforts. The main objective is to develop a risk based approach for computing and monitoring the security features of interactively complex software-reliant systems across the lifespan and supply chain. To accomplish this goal Mission Risk Analytic (MRA) is developed.

Keywords: Risk analysis, Software Engineering, Software Security, Complex Systems, Mission risk analytic (MRA)

I. INTRODUCTION

Several organizations measure just for the sake of measuring, with minute or no thought given to what purpose and business purposes are being satisfied or what questions each measure is intended to answer. However, meaningful measurement is about transforming strategic direction, policy, and other forms of management decision into action and measuring the performance of that action. Effective measures express the extent to which objectives are being met, how well requirements are being satisfied, how well processes and controls are functioning, and the extent to which performance outcomes are being achieved. The basic goal of measurement and analysis is to provide decision makers with the information they need, when they need it, and in the right form. In recent years, researchers have begun to turn their attention to the topic of software security assurance and how to measure it. Software security assurance is justified confidence that software-reliant systems are adequately planned, acquired, built, and fielded with sufficient security to meet operational needs, even in the presence of attacks, failures, accidents, and unexpected events. For several years, various groups within the software engineering community have been working diligently to identify practices aimed at developing more secure software. However, efforts to measure software security assurance have yet to materialize in any substantive fashion, although some foundational work has been performed. Program's core competency is in software and information security, software engineering measurement and analysis. The purpose of this new research project is to address the following two questions:

a. How do we establish, specify, and measure justified confidence that interactively complex software-reliant systems are sufficiently secure to meet operational needs?
b. How do we measure at each phase of the development or acquisition life cycle that the required/desired level of security has been achieved?

In essence, the two research questions examine how decision makers (for example, development program and project managers as well as acquisition program officers) can measure and monitor the security characteristics of interactively complex software-reliant systems across the life cycle and supply chain.

II. RISK ANALYSIS

Risk analysis is a management tool, the standards for which are determined by whatever management decides it is willing to accept in terms of actual loss. It is used to identify and assess factors that may jeopardize the success of a project or achieving a goal. This also helps to define preventive measures to reduce the probability of these factors from occurring and identify countermeasures to successfully deal with these constraints when they develop to avert possible negative effects on the competitiveness of the company. It can provide management with vital information on which to base sound Decisions. Thorough risk analysis can provide answers to many questions, such as:

Is it always best to prevent the occurrence of a situation? Is it always possible? Should policy also consider how to contain the effect a hazardous situation may have?

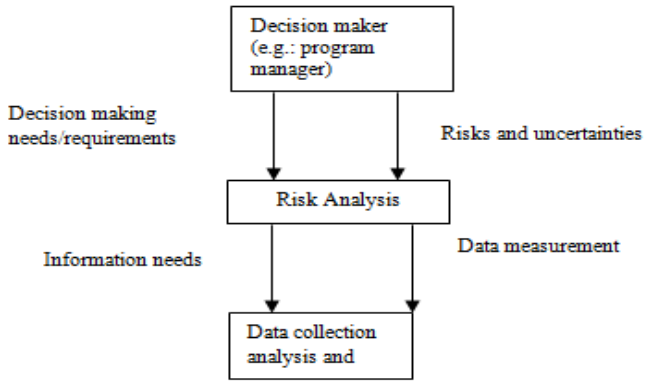

Figure.1 Risk-Based Decision Making

CONFERENCE PAPER
Two day National Conference on Advanced Trends and Challenges in Computer Science and Applications
Organized by: Shree Vishnu Engineering College for Women, Bhimavaram A.P.
Schedule: 18-19 March 2014

A. Two Approaches for Analysing Risk:

Our research is focused on developing risk-based approaches for measuring and analysing the performance of interactively complex software-reliant systems across the life cycle and supply chain. To fully appreciate what this statement means, you need to understand the phrase, "interactively complex software-reliant systems." A socio-technical system is defined as interrelated technical and social elements that are engaged in goal-oriented behaviour.

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. Projects, programs, and operational processes are all examples of socio technical systems. A software-reliant system is a socio-technical system whose behaviour (e.g., functionality, performance, safety, security, interoperability, and so forth) is dependent on software in some significant way. In the remainder of this document, when we use the word system, we are referring to a software-reliant system. Interactive complexity refers to the presence of unplanned and unexpected sequences of events in a system that are either not visible or not immediately understood. The components in an interactively complex system interact in relatively unconstrained ways. When a system is interactively complex, independent failures can interact with the system in ways that cannot be anticipated by the people who design and operate the system. Measurement and analysis should be tailored to the context in which it will be applied. In our research project, we have been focused on using risk analysis to direct the measurement and analysis of interactively complex systems. Two distinct risk analysis approaches can be used when evaluating systems: (1) tactical risk analysis and (2) systemic risk analysis.

III. TACTICAL RISK ANALYSIS

Risk is the probability of suffering harm or loss. From the tactical perspective, risk is defined as the probability that an event will lead to a negative consequence or loss. The basic goal of tactical risk analysis is to evaluate a system's components for potential failures. Tactical risk analysis is based on the principle of system decomposition and component analysis. The first step of this approach is to decompose a system into its constituent components. The individual components are then prioritized, and a subset of components is designated as being critical. Next, the risks to each critical component are analysed. Tactical risk analysis enables stakeholders to (1) determine which components are most critical to a system and (2) Analyse ways in which those critical components might fail (i.e., analyse the risk to critical components). Stakeholders can then implement effective controls designed to mitigate those potential failures. Because of its focus on preventing potential failures, tactical risk analysis has been applied extensively within the discipline of systems engineering. However, analysts need to understand the limitations of using tactical risk analysis to evaluate interactively complex systems, which include the following:

a. Only critical components are analysed. Non-critical components are not examined, and Inter dependencies among components are not addressed.
b. The selection of which conditions and events (i.e., sources or causes of risk) to consider is subjective.
c. Non-linear relationships among conditions and events (e.g., feedback) are not considered. Risk causal relationships are presumed to be simple, direct, and linear.
d. Events that produce extreme or catastrophic consequences are difficult to predict because they can be triggered by the contemporaneous occurrences of multiple events, cascading consequences, and emergent system behaviours.
e. Confidence in the performance of individual components does not establish confidence in the performance of the parent system.

In addition, when you attempt to decompose interactively complex systems, some system-wide behaviours become lost. It is very difficult to establish the relationship between the macro-level behaviour of the system and the micro-level behaviour of individual components. As a result, tactical risk analysis provides a partial picture of the risks to an interactively complex system. To get a more holistic view of risk in an interactively complex system, you need to employ an alternative analysis approaches

IV. SYSTEMIC RISK ANALYSIS

The Mission Risk Analytic (MRA) is developed to enable systemic risk analysis of interactively complex systems. During our research and development activities over the past few years, we demonstrated how the MRA provides an efficient and effective means of analysing risk in interactively complex systems, such as acquisition programs. Our current work builds on this initial research by exploring how to apply the MRA in a software security context. When the MRA is applied in this context, the target of the risk analysis is the project6 or a program7 that is acquiring and developing a software product (e.g., an integrated software reliant system).The goal is to gauge whether the security risk of the deployed software product will be within an acceptable tolerance. The following two tasks form the foundation of the MRA: (1) driver identification and (2) driver analysis. This section describes both tasks in detail, presents a discussion of the driver profile, which is the main output of the MRA, introduces the concept of mission risk, and concludes with a description of the MRA's key tasks and steps. The concepts and examples presented in this section are described in the context of a large-scale acquisition and development program, which is one specific type of interactively complex system.

A. Driver Identification:

The main goal of driver identification is to establish a set of factors, called drivers, that can be used to measure performance in relation to a program's mission and objectives. Once the set of drivers is established, analysts can then evaluate each driver in the set to gain insight into the likelihood of achieving the mission and objectives. To measure performance effectively, analysts must ensure that the set of drivers conveys sufficient information about the mission and objectives being evaluated. As a result, the first step in identifying a set of drivers is to establish the mission.

CONFERENCE PAPER
Two day National Conference on Advanced Trends and Challenges in Computer Science and Applications
Organized by: Shree Vishnu Engineering College for Women, Bhimavaram A.P.
Schedule: 18-19 March 2014

B. Mission risk analytic (MRA):

From the systemic perspective, risk is defined as the probability of mission failure (i.e., not achieving key objectives). Systemic risk, also referred to as mission risk in this document, examines the aggregate effects of multiple conditions and events on a system's ability to achieve its mission. Systemic risk analysis is based on system theory. The underlying principle of system theory is to analyse a system as a whole rather than decomposing it into individual components and then analysing each component separately. In fact, some properties of a system are best analysed by considering the entire system, including

a. Influences of environmental factors
b. Feedback and nonlinearity among causal factors
c. Systemic causes of failure (as opposed to proximate causes)
d. Emergent properties

Systemic risk analysis thus provides a holistic view of the risk to an interactively complex socio technical system. The first step in this type of risk analysis is to establish the objectives that must be achieved. The objectives define the desired outcome, or "picture of success," for a system. Next, systemic factors that have a strong influence on the outcome (i.e., whether or not the objectives will be achieved) are identified. These systemic factors, called *drivers* in this report, are important because they define a small set of factors that can be used to assess a system's performance and gauge whether it is on track to achieve its key objectives. The drivers are then analysed, which enables decision makers to gauge the overall risk to the system's mission.

C. Mission & Objectives:

The MRA defines the term *mission* as the fundamental purpose of the system that is being examined. In the context of an acquisition program, the mission can be expressed in terms of the software product that is being acquired, developed, and deployed. The mission statement is important because it defines the target, or focus, of the analysis effort. After the basic target has been established, the next step is to identify which specific aspects of the mission need to be analysed in detail. In the MRA, an *objective* is defined as a tangible outcome or result that must be achieved when pursuing a mission. Each mission typically comprises multiple objectives. The goal of the second step of driver identification is to determine which of those objectives will be assessed. Selecting objectives refines the scope of the assessment to address specific aspects of the mission that are important to decision makers. In general, objectives identified during the MRA should meet the following criteria:

a. ***Specific—*** The objective is concrete, detailed, focused, and well defined. It emphasizes action and states a specific outcome to be accomplished.
b. ***Measurable—*** The objective can be measured, and the measurement source is identified.
c. ***Achievable—*** The expectation of what will be accomplished is attainable given the time period, resources available, and so on.
d. ***Relevant—*** The outcome or result embodied in the objective supports the broader mission being pursued.
e. ***Time-bound—*** The timeframe in which the objective will be achieved is specified.

During driver identification, analysts must select one or more objectives that will be analyzed. The number of objectives depends on the breadth and nature of the issues being investigated. The following is an example of a generic objective for determining whether an acquisition program is adequately addressing software security: When the system is deployed, security risks to the deployed system will be within an acceptable tolerance. This example is fairly abstract; additional details must be added to the objective to meet the criteria listed above.

(a). which system is being deployed
(b). when that system is expected to be deployed
(c). how risk will be measured
(d). how "acceptable tolerance" is defined for the program

The field experience shows that many decision makers (e.g., acquisition program managers) have difficulty constructing objectives that meet the above criteria for objectives. While decision makers have a tacit understanding of their objectives, they often cannot precisely articulate or express the objectives in a way that addresses the criteria. If the program's objectives are not clearly articulated, decision makers can have trouble assessing whether the program is on track for success. To address this issue, qualitative implementations of the MRA allow for imprecise expressions of objectives. Specific information about objectives that is tacitly understood by program managers and staff becomes more explicit during execution of the MRA. The remainder of this section describes a qualitative implementation of the MRA. We are also working on quantitative implementation of the MRA, which we intend to present in other reports.

D. Drivers:

The MRA defines a driver as a factor that has a strong influence on the eventual outcome or result (i.e., whether or not objectives will be achieved). Table 1 highlights three key attributes of a driver: name, success state, and failure state. The example driver in the table is named Security Process, and it examines how the program's processes are affecting achievement of the software security objective. Table 1 also indicates that each driver has two possible states: a success state and a failure state. The success state means that the program's processes incorporate security considerations adequately, which helps enable the achievement of the objectives. In contrast, the failure state signifies that the program's processes do not adequately incorporate security considerations and, as a result, the objectives will not be achieved.

Table 1 Analytic States

Attribute	***Description***	***Example***
Name	A concise label that describes the basic nature of the driver.	Security process
Success state	A driver exerts a positive influence on the outcome	The process being used to develop and deploy the system sufficiently incorporates security.
Failure state	A driver exerts a negative influence on the outcome	The process being used to develop and deploy the system does not sufficiently incorporate security.

Analysis of a driver requires determining how it is currently acting (i.e., its current state) by examining the effects of conditions and potential events on that driver. The goal is to determine if the driver is

(a). almost certainly in its success state
(b). most likely in its success state
(c). equally likely to be in its success or failure states
(d). most likely in its failure state
(e). almost certainly in its failure state

The above list can be used define a qualitative scale for driver analysis. Analysing each driver in relation to the qualitative scale establishes a benchmark of performance in relation to a system's documented mission and objectives.

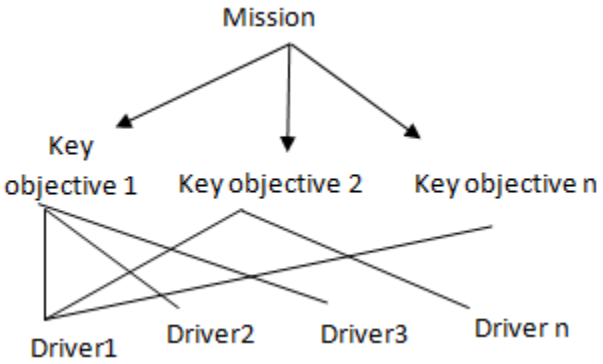

Figure 2: Relationships among Objectives and Drivers

E. Deriving a Set of Drivers:

The starting point for identifying a set of drivers is to articulate the mission and objectives that are being assessed. Analysts can then derive a set of drivers from them. The relationships among mission, objectives, and drivers are depicted in Figure 3. When dealing with multiple objectives, analysts must be sure to record these relationships to enable effective decision making.

Deriving a unique set of drivers based on the program's mission and objectives requires gathering information from people with experience and expertise relevant to the specified mission and objectives. For example, identifying a set of drivers for software development objectives requires input from acquisition programs managers and software-reliant systems developers. Similarly, analysts seeking to identify a set of drivers for software security would consult with security experts.

The experts from whom information is elicited should be familiar with the objectives that have been defined. Analysts can use the objectives to focus interviews or discussions with experts. During interviews or discussions, experts answer the following questions:

a. What circumstances, conditions, and events will drive your program toward a *successful* outcome?
b. What circumstances, conditions, and events will drive your program toward a *failed* outcome?

After they obtain information from the experts, analysts organize the information into approximately 10–25 groups that share the driver as the central idea or theme of each group. SEI staff has employed this approach for identifying drivers in a variety of areas, including software acquisition and development programs, cyber security processes, and business portfolio management. The most recent focus has been on establishing drivers for software security. The next section presents a set of software security drivers.

V. REFERENCES

[1] K.V.D.Kiran and Dr.L.S.S.Reddy, "A Comparative Analysis on Risk Assessment Information Security Models," International Journal of Ccomputer Applications" , vol. 82, pp.41-46, November 2013.

[2] K.V.D.Kiran and Dr.L.S.S.Reddy, "A Critical study on information security Risk assessment using Fuzzy and Entropy Methodologies "International Journal of Computers and Communications, vol 1,pp.17-21,December 2012.

[3] 1.Park, Robert; Goethert, Wolfhart; & Florac, William. Goal-Driven Software Measurement —A Guidebook (CMU/SEI-96-HB-002). Software Engineering Institute, Carnegie Mellon University, 1996.

[4] Perrow, Charles. Normal Accidents: Living with High-Risk Technologies. Princeton University Press, 1999.

[5] Software Engineering Institute. Measurement & Analysis. http://www.sei.cmu.edu/measurement (2010)

.

.