Fredrik Seehusen
Michael Felderer
Jürgen Großmann
Marc-Florian Wendland (Eds.)

LNCS 9488

Risk Assessment and Risk-Driven Testing

Third International Workshop, RISK 2015
Berlin, Germany, June 15, 2015
Revised Selected Papers

Lecture Notes in Computer Science **9488**

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

More information about this series at http://www.springer.com/series/7408

Fredrik Seehusen · Michael Felderer
Jürgen Großmann · Marc-Florian Wendland (Eds.)

Risk Assessment and Risk-Driven Testing

Third International Workshop, RISK 2015
Berlin, Germany, June 15, 2015
Revised Selected Papers

Springer

Editors
Fredrik Seehusen
SINTEF ICT
Oslo
Norway

Jürgen Großmann
Fraunhofer Institut FOKUS
Berlin
Germany

Michael Felderer
Institut für Informatik
Universität Innsbruck
Innsbruck
Austria

Marc-Florian Wendland
Fraunhofer Institut FOKUS
Berlin
Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-26415-8 ISBN 978-3-319-26416-5 (eBook)
DOI 10.1007/978-3-319-26416-5

Library of Congress Control Number: 2015953793

LNCS Sublibrary: SL2 – Programming and Software Engineering

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

The continuous rise of software complexity with increased functionality and accessibility of software and electronic components leads to an ever-growing demand for techniques to ensure software quality, dependability, safety, and security. The risk that software systems do not meet their intended level of quality can have a severe impact on vendors, customers, and even — when it comes to critical systems and infrastructures — daily life. The precise understanding of risks, as well as the focused treatment of risks, has become one of the cornerstones for critical decision making within complex social and technical environments. A systematic and capable risk and quality assessment program and its tight integration within the software development life cycle are key to building and maintaining secure and dependable software-based infrastructures.

This volume contains the proceedings of the Third International Workshop on Risk Assessment and Risk-Driven Testing (RISK 2015) held in June 2015 in Berlin, Germany, in conjunction with the OMG Technical Meeting, June 15–19, 2015. The workshop brought together researchers from the European Union to address systematic approaches combining risk assessment and testing. During the workshop, eight peer-reviewed papers were presented and actively discussed. The workshop was structured into three sessions namely: "Risk Assessment," "Risk and Development," and "Security Testing." The program was completed by a keynote on "Fundamental Principles of Safety Assurance" from Prof. Tim Kelly.

Owing to its integration with the OMG Technical Meeting, the workshop initiated a fruitful discussion between participants from industry and academia.

We would like to take this opportunity to thank the people who contributed to the RISK 2015 workshop. We want to thank all authors and reviewers for their valuable contributions, and we wish them a successful continuation of their work in this area.

September 2015

Jürgen Großmann
Marc-Florian Wendland
Fredrik Seehusen
Michael Felderer

Organization

RISK 2015 was organized by Fraunhofer FOKUS, SINTEF ICT, and the University of Innsbruck.

Organizing Committee

Jürgen Großmann	Fraunhofer FOKUS, Germany
Marc-Florian Wendland	Fraunhofer FOKUS, Germany
Fredrik Seehusen	SINTEF ICT, Norway
Michael Felderer	University of Innsbruck, Austria

Program Committee

Fredrik Seehusen	SINTEF ICT, Norway
Michael Felderer	University of Innsbruck, Austria
Jürgen Großmann	Fraunhofer FOKUS, Germany
Marc-Florian Wendland	Fraunhofer FOKUS, Germany
Ina Schieferdecker	FU Berlin/Fraunhofer FOKUS, Germany
Ketil Stølen	SINTEF ICT, Norway
Ruth Breu	University of Innsbruck, Austria
Ron Kenett	KPA Ltd. and University of Turin, Italy
Sardar Muhammad Sulaman	Lund University, Sweden
Bruno Legeard	University of Franche-Comté, France
Gabriella Carrozza	SELEX ES, Italy
Shukat Ali	Simula Research Laboratory, Norway
Markus Schacher	KnowGravity Inc., Switzerland
Alessandra Bagnato	Softeam, France
Kenji Taguchi	AIST, Japan
Zhen Ru Dai	University of Applied Science Hamburg, Germany
Tim Kelly	University of York, UK

Contents

Risk Assessment

Risk Assessment and Security Testing of Large Scale Networked Systems with RACOMAT

Johannes Viehmann[1(✉)] and Frank Werner[2]

[1] Fraunhofer FOKUS, Berlin, Germany
Johannes.Viehmann@fokus.fraunhofer.de
[2] Software AG, Darmstadt, Germany
Frank.Werner@softwareag.com

Abstract. Risk management is an important part of the software quality management because security issues can result in big economical losses and even worse legal consequences. While risk assessment as the base for any risk treatment is widely regarded to be important, doing a risk assessment itself remains a challenge especially for complex large scaled networked systems. This paper presents an ongoing case study in which such a system is assessed. In order to deal with the challenges from that case study, the RACOMAT method and the RACOMAT tool for compositional risk assessment closely combined with security testing and incident simulation for have been developed with the goal to reach a new level of automation results in risk assessment.

Keywords: Risk assessment · Security testing · Incident simulation

1 Introduction

For software vendors risk assessment is a big challenge due to the steadily increasing complexity of today's industrial software development and rising risk awareness on the customer side. Typically, IT systems and software applications are distributed logically and geographically, and encompass hundreds of installations, servers, and processing nodes. As customers rely on mature and ready-to-use software, products should not expose vulnerabilities, but reflect the state of the art technology and obey security risks or technical risks. Failing to meet customer expectations will result in a loss of customer trust, customer exodus, financial losses, and in many cases in legal consequences and law suits.

On the other hand, the impossibility to analyze and treat every potential security problem in advance is well-known. Any security issue without appropriate safeguards could lead to a considerable damage for the customer, be it its loss of business (e.g. when a successful DoS attack prevents business processes form being pursued), loss of data (due to unauthorized access) or malicious manipulation of business process sequences or activities. The task of risk management is to identify and treat the most critical risks without wasting resources for less severe problems. Within this paper, only the risk assessment part of the risk management process is addressed. More precisely, this paper reports the experiences made during the risk assessment for an industrial large scale software system Command Central.

F. Seehusen et al. (Eds.): RISK 2015, LNCS 9488, pp. 3–17, 2015.
DOI: 10.1007/978-3-319-26416-5_1

Risk assessment can be difficult and expensive. It typically depends on the skills and estimates of experts and manual risk assessment can only be performed at a high level of abstraction for large scale systems. Security testing is one risk analysis method that eventually yields objective results. But security testing itself might be hard and expensive, too. Manual testing is itself error prone and again infeasible for large scale systems. Choosing what should be tested and interpreting security test results are not trivial tasks. Indeed, even highly insecure systems can produce lots of correct test verdicts if the "wrong" test cases have been created and executed. Therefore, it makes sense to do Risk Assessment COMbined with Automated Testing, i.e. to use the RACOMAT method and the RACOMAT tool introduced here. RACOMAT has been developed along the case study in order to deal exactly with the challenges of large scale networked systems. Both, the development of RACOMAT and the risk assessment of Command Central are still ongoing.

1.1 The Case Study

The software under analysis is called Command Central [14] from Software AG, a tool from the webMethods tool suite allowing release managers, infrastructure engineers, system administrators, and operators to perform administrative tasks from a single location. Command Central assist the configuration, management, and monitoring by supporting the following tasks:

- Infrastructure engineers can see at a glance which products and fixes are installed, where they are installed, and compare installations to find discrepancies.
- System administrators can configure environments by using a single web user interface or command-line tool. Maintenance involves minimum effort and risk.
- Release managers can prepare and deploy changes to multiple servers using command-line scripting for simpler, safer lifecycle management.
- Operators can monitor server status and health, as well as start and stop servers from a single location. They can also configure alerts to be sent to them in case of unplanned outages.

Command Central is built on top of Software AG Common Platform, which uses the OSGi (Open Services Gateway Initiative) framework. Product-specific features are in the form of plug-ins.

Command Central users can communicate with Command Central Server using either the graphical web user interface for administering products using the web, or the Command line interface for automating administrative operations. An architecture overview of the Command Central software is provided in Fig. 1.

The Command Central Server accepts administrative commands that users submit through one of the user interfaces and directs the commands to the respective Platform Manager for subsequent execution. An installation in Command Central means one or more instances of the products that Command Central can manage. It provides a common location for configuring managed products installed in different environments.

The webMethods Platform Manager manages other Software AG products. Platform Manager enables Command Central to centrally administer the lifecycle of

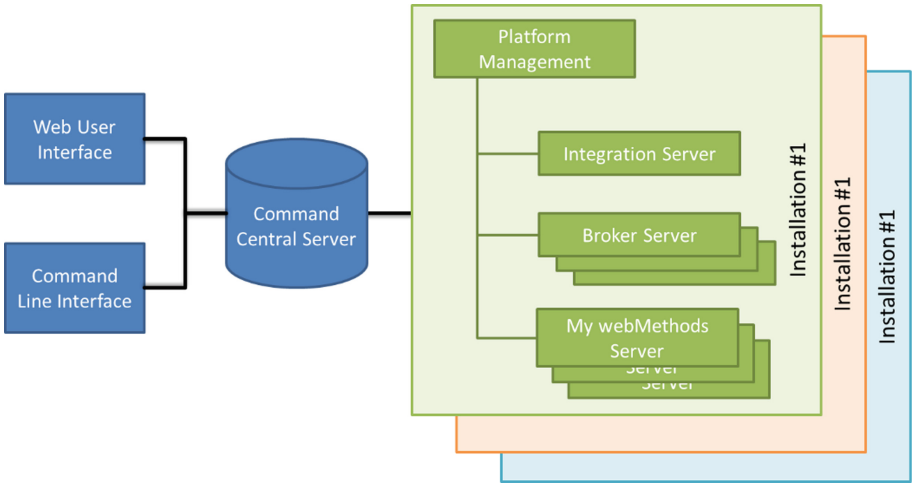

Fig. 1. An installation set-up scenario with command central management

managed products. In a host machine, there might be multiple Software AG product installations. For each Software AG product installation, a separate Platform Manager is needed to manage the installed products.

2 State of the Art

Security critical ICT systems should be carefully managed especially with respect to the related security risks. Such a risk management should include well-known concepts like risk assessment (ISO 31000 – [2]) and security testing (ISO 29119 – [3]).

2.1 Risk Assessment

According to the ISO 31000 standard, risk assessment means to identify, analyze and evaluate risks which could damage or destroy assets [2]. Lots of different methods and technologies for risk assessment have evolved, including fault tree analysis (FTA) [5], event tree analysis ETA [6], Failure Mode Effect (and Criticality) Analysis FMEA/FMECA [4] and the CORAS method [1].

Compositional risk assessment allows analysts to deal with manageable small components of a complex large scale modular system. It combines the individual risk assessment results for components to derive a risk picture for the entire complex system without looking further into the details of its components. However, most traditional risk assessment technologies analyze systems as a whole [7]. They do not offer support for compositional risk assessment. Nevertheless there are some publications dealing with compositional risk assessment and suggesting extensions for the mentioned risk assessment concepts, e.g. [8] for FTA and [9] for FMEA or [10] for CORAS, which is used in the case study presented here.

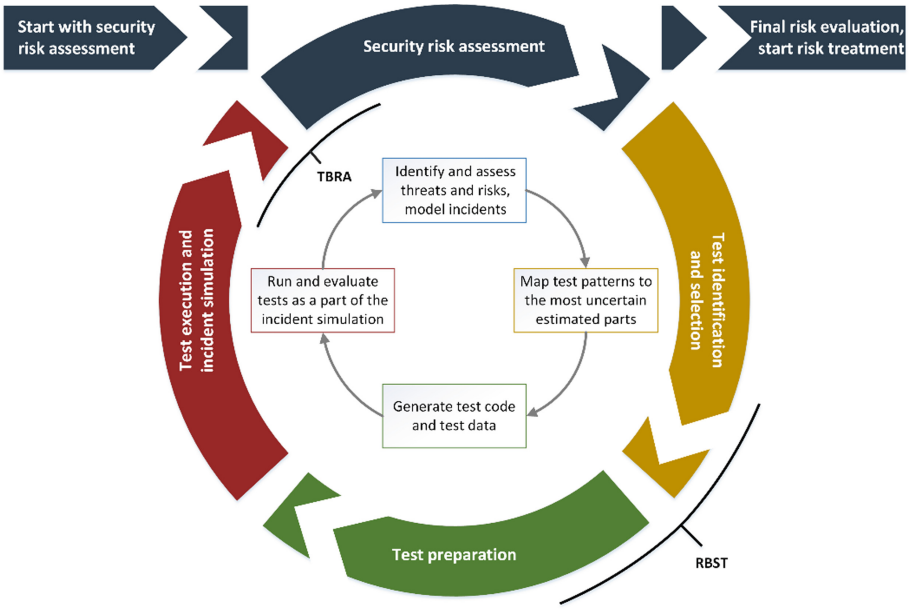

Fig. 2. The RACOMAT method

There are huge databases of common weaknesses, attack patterns and safeguards available that can be used as a base for security risk assessment, for example Mitre CWE [23] and CAPEC [22] or BSI IT-Grundschutz [21]. There are also vulnerability databases that list specific vulnerabilities for existing software programs, e.g. Mitre CVE [24]. Such information could be helpful in compositional risk assessment for systems that use listed programs or that have them in their environment. The mentioned catalogues are used in the case study introduced here.

2.2 Security Testing and Testing Combined with Risk Assessment

The ISO 29119 standard defines security testing as a type of testing that tries to evaluate the protection level of the system under test against unauthorized access, unwanted use and denial of service [3].

Traditional testing is a method to analyze the behavior of a system according to its specified functionality and expected results. Security testing in contrast also tests for unspecified behavior and for unexpected results. Hence, compared to other types of testing, for security testing it is harder to decide what should actually be tested and is more challenging to interpret the observed behavior.

One possible way to deal with the challenges of security testing is to combine it with risk assessment. ISO 29119 defines risk-based testing as a general method that uses risk assessment results to guide and to improve the testing process [3]. Risk analysis results can be used to define test policies and strategies, to identify what should actually be tested and how much effort should be spend for it. For instance, Kloos et al.

use fault trees for identifying test cases [15]. Stallbaum and Metzger automated the generation and prioritization of test cases based upon risk assessment artefacts [16]. Even (semi-) automated risk-based security testing might be expensive. Reusing testing artefacts for recurring security testing problems might help to reduce the effort. Security test patterns have been suggested for that purpose [17], but currently there is no extensive library of useful security test patterns available.

Risk assessment and testing can also interact the other way around: in test-based risk assessment, test results are used to identify, analyze and evaluate risks. There are several publications about this approach [18, 19], but there is still no general applicable method and not much tool support.

The concepts of risk-based testing and test-based risk assessment can also be combined. While Erdogan et al. propose such a combination [18], they do not propose any technique or detailed guideline for how to update the risk model based on the test results. The combined method with tool support presented in [20] has been developed further along the case study presented here and it was named the RACOMAT method and tool.

2.3 Simulation

Simulations that work with simplifying models in general can be very helpful to analyze large scale systems. For instance, Monte Carlo simulations can be used to analyze the behavior of complex systems and especially for calculating likelihood values in the risk aggregation process [11, 12].

In [13] it is described how Monte Carlo simulations and security testing can be used together within an iterative risk assessment process in order to refine the risk picture. This approach is used in the case study described here.

3 Requirements, Problems and Expectations

Large scale networked computer systems and software programs can be enormous complex. Building and maintaining such systems is challenging and it can be very expensive. One way to reduce costs without reducing the product features and the product quality is to let as much work as possible be done automatically by machines.

However, in the production and lifecycle management process for software there is usually only limited potential for automation. It typically requires lots of creative work which nowadays has to be done manually, e.g. modelling or writing code directly.

Nevertheless, at least for analytical and recurring tasks in the software development and maintenance process, a high level of automation should be achievable.

These promising candidates for automation include especially testing as a vital part of the software quality management. Indeed, within a testing process, many tasks can be automated, especially test data generation and test case execution. There are lots of tools supporting automated testing. Testing for specified behavior can be more or less completely automated if the specification is well modeled – test cases can be derived automatically from such a model. Of course, appropriate models have to be created