Risk-driven Security Testing versus Test-driven Security Risk Analysis *

Gencer Erdogan[1,2]
Supervised by: Ketil Stølen[1,2]

[1] Department of Informatics, University of Oslo, PO Box 1080 Blindern, N-0316 Oslo, Norway
[2] Department for Networked Systems and Services, SINTEF ICT, PO Box 124 Blindern, N-0314 Oslo, Norway
{gencer.erdogan,ketil.stolen}@sintef.no

Abstract. It is important to clearly distinguish the combinations of security testing and security risk analysis depending on whether it is viewed from a security testing perspective or a security risk analysis perspective. The main focus in the former view is security testing in which test objectives are to be achieved, while the main focus in the latter view is security risk analysis with the aim to fulfill risk acceptance criteria. The literature's lack of addressing this distinction is accompanied with the lack of addressing two immediate problems within this context, namely the gap between high-level security risk analysis models and low-level security test cases, and the consideration of investable effort. We present initial ideas for methods that address these problems followed by an industrial case study evaluation in which we have gathered interesting results.

Keywords: Security risk analysis, Security testing

1 Introduction

Security testing is a process to determine that an information system protects data and maintains functionality as intended [1]. Security risk analysis is a specialized risk analysis approach in which information security risk is associated with the potential that threats will exploit vulnerabilities of an information asset, or group of information assets, and thereby cause harm to an organization [2].

The literature collectively refers to the combinations of security testing and security risk analysis as risk-based security testing. It is, however, important to make a clear distinction between such combinations depending on whether it is viewed from (A) a security testing perspective, or (B) a security risk analysis perspective. In (A) the main focus is security testing in which test objectives are to be achieved while risk analysis is used as a means to make security testing

* This work has been conducted as a part of the DIAMONDS (201579/S10) project funded by the Research Council of Norway, as well as a part of the NESSoS network of excellence funded by the European Commission within the 7th Framework Programme.

more effective. We name this approach Risk-driven Security Testing (RST). In (B) the main focus is security risk analysis with the aim to fulfill risk acceptance criteria while security testing is used as a means to develop and/or validate risk models. We name this approach Test-driven Security Risk Analysis (TSR). Furthermore, we address two key challenges within both RST and TSR. The first challenge is to bridge the gap between high-level security risk models and low-level security test cases, while the second challenge is to make sure that the investable effort is correctly reflected in RST and TSR. As an initial evaluation of our proposed methods for RST and TSR, we have carried out an industrial case study evaluation of a TSR based approach in which we have gathered interesting results.

The paper is structured as follows: Sect. 2 outlines the research objectives followed by a brief description of the research method, Sect. 3 gives an overview of the work done to date, Sect. 4 presents preliminary results from an industrial case study in which a TSR approach was tried out, and Sect. 5 concludes the paper.

2 Research Objectives and Research Method

Security risk analysis models are often at a high-level of abstraction (e.g. business level), while security test cases are at a low-level of abstraction (e.g. implementation level), and the challenge in RST is to identify the most important security test cases, while the challenge in TSR is to identify an accurate security risk model of the target of evaluation.

These challenges are important to address in order to define exactly what to test, produce only the necessary security test cases and to obtain an accurate security risk model. The two first points provide the security testers with an indication for when to terminate the security testing process, i.e. to limit the scope of the security testing process. The necessity to limit the scope of the security testing process is due to the fact that security is often constrained by cost and time – one example from the industry is the author's personal experience of conducting security testing in a European organization [3]. It is therefore essential to take the effort available for conducting an RST or a TSR into account.

The author seeks to address the particular task of developing an industrial guideline for effort dependent risk-driven security testing and test-driven security risk analysis. The initial research question is the following:

– What is a good industrial guideline for effort dependent risk-driven security testing and test-driven security risk analysis?

The research work will adopt an iterative incremental approach where the industrial partners provide industrial case studies on RST and TSR. There will, in total, be carried out six case studies. In this light, the research process will be conducted using the Technology Research Method [4] which is an iterative incremental research method.

3 Current Work

Figure 1 presents the steps in RST and TSR. The steps for security testing are in line with the steps presented in the Standard for Software Component Testing [6], which is also one of the building blocks in the upcoming new international standard for software testing - ISO/IEC 29119 [7]. The steps for security risk analysis are in line with the steps presented in ISO 31000 [8].

In **RST** security testing is supported by security risk assessment in order to make security testing more effective. The aim is to focus the security testing process to carry out security tests on the most important parts of the system under test, and to execute only the most important security test cases. RST provides two alternatives to achieve this aim. **The first alternative** (Step 3) achieves this by first directing the identified security test model from Step 2 as input to Step 3. Based on this input, risks concerning the system under test are identified, estimated and evaluated in Step 3. The output of Step 3 is a risk evaluation matrix that contains a list of prioritized risks of the system under test. The risk evaluation matrix is finally used as a basis for generating and prioritizing security test cases in Step 4. **The second alternative** (Step 5) achieves this by first directing the identified security test cases from Step 4 as input to Step 5. Based on this input, risks addressed by the security test cases are identified, estimated and evaluated in Step 5. The output of Step 5 is a risk evaluation matrix that contains a list of prioritized risks of the system under test. The risk evaluation matrix is finally used as a basis for eliciting the most important security test cases to be executed in Step 6. The dashed rectangles that surrounds Step 3 and 5 indicate that the security testing process can be supported by either Step 3 *or* Step 5 when conducting the RST method.

In **TSR** security risk analysis is supported by security testing in order to develop and/or validate risk models. The aim is to strengthen the correctness of the security risk analysis models. TSR provides two alternatives to achieve this aim. **The first alternative** (Step 2 and 3) supports the *development* of risk models by identifying potential risks. This is achieved by first directing the system model and the identified assets of the target of evaluation from Step 1 as input to Step 2. Based on this input, security test cases are generated and prioritized in Step 2. The identified security test cases are then executed in Step 3. Finally, the security testing results are used as a basis for identifying potential risks which are used as an input to Step 4. **The second alternative** (Step 5 and 6) supports the *validation* of risk models by first identifying security tests that explore the risks and then validating the risk model based on the security testing results. This is achieved by first directing the risk evaluation matrix from Step 4 as input to Step 5. Based on this input, security test cases are generated and prioritized in Step 5. The identified security test cases are then executed in Step 6. Finally, the security testing results are used as a basis for validating and updating the security risk analysis models. The dashed rectangles that surrounds Step 2 and 3 and Step 5 and 6 indicate that the security risk analysis process can be supported by security testing for either developing risk models or validating risk models, *or both* when conducting the TSR method.

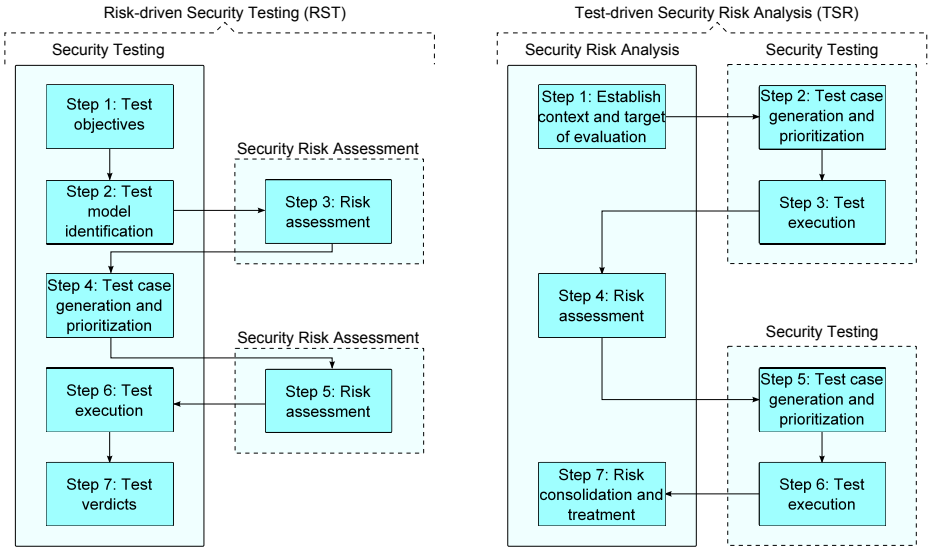

Fig. 1. *The steps in Risk-driven Security Testing and Test-driven Security Risk Analysis.*

4 Case Study

We have carried out an industrial case study using a TSR based approach. In particular, we carried out the alternative for *validating* the risk model, i.e. step 1, 4, 5, 6 and 7 of the TSR approach, as explained in Sect. 3. The target system analyzed was a multilingual, web-based e-business application, which serves as the backbone for the system owner's business goals and is used by a large amount of users every day. The case study was carried out in a period of four months in form of meetings and security testing sessions. The presented results are limited to the experiences obtained in the case study due to confidentiality reasons.

Table 1 outlines the process undergone during the case study. The first column specifies the meeting sequence (SRA denotes a security risk analysis meeting, while ST denotes a security testing meeting). The second column lists the participants (C denotes participants from the customer organization, while A denotes participants from the analysis team). The third column describes the contents and achievements in each meeting. Finally, the fourth column shows the approximate time spent (in man-hours) for each meeting. The time spent on work before and after the meetings is not included in the table.

Security risk analysis was conducted using the CORAS approach [5]. The process of identifying security tests was carried out by first using the risk evaluation matrix with identified risks as a starting point. Then, the threat scenarios that lead up to a risk that needed to be treated were systematically identified as *testable* or *not testable*. Furthermore, the identified testable threat scenarios were prioritized based on their individual effort for realizing the risk. Finally, secu-

Table 1. The process undergone during the case study

Meeting	Participants	Contents	Hours
1 - SRA	**C**:One domain expert. **A**:The analyst. Two domain experts.	Defining the goals, context, target, focus and scope of the SRA.	2
2 - SRA	**C**:One domain expert. One developer. **A**:The analyst. The secretary. One domain expert.	Defining the goals, context, target, focus and scope of the SRA.	3
3 - SRA	**C**:One domain expert. **A**:The analyst. The secretary.	Concretizing the scope, assets and risk evaluation criteria of the SRA.	6
4 - SRA	**C**:One domain expert. **A**:The analyst. The secretary.	Identifying and evaluating risks.	6
5 - SRA / ST	**C**:One domain expert. **A**:The analyst. The secretary.	Identifying security tests.	6
6 - ST	**A**:The analyst. The secretary.	Implementing security tests.	8
7 - ST	**A**:The analyst. The secretary.	Executing security tests.	6
8 - ST	**A**:The analyst. The secretary.	Executing security tests.	6
9 - SRA	**C**:One domain expert. One developer. **A**:The analyst. The secretary.	Validating and updating the risk model based on the security testing results. Suggesting treatments for security tests that failed.	2

rity tests were identified for the prioritized testable threat scenarios. A testable threat scenario, in this context, is simply a threat scenario that is possible to test at the software level.

Table 2 presents the overall TSR results obtained in the case study. The leftmost column specifies the information security assets taken into consideration during the security risk analysis process. The row named *Common* denotes the number of security risk analysis elements that are common for all assets. The topmost row specifies the number of security risk analysis elements: R denotes the number of risks, TS denotes the number of threat scenarios, RT denotes the number of tested risks, TST denotes the number of tested threat scenarios (the tested threat scenarios that addressed confidentiality also addressed integrity and are therefore not counted two times in the total sum), R *Upd.* denotes the number of updated risks based on the security testing results and TS *Upd.* denotes the number of updated threat scenarios based on the security testing results.

A total number of 31 risks and 43 threat scenarios were identified during the risk assessment, and from these, 11 risks and 7 threat scenarios were tested. Approximately 80% of the identified security tests that explored these risks and threat scenarios uncovered some form of security vulnerability. This is an indication that the risk model contributed to identify relevant security tests. Furthermore, the security testing results helped us to validate and update the risk model; approximately 20% of all risks and 14% of all threat scenarios had

Table 2. Test-driven Security Risk Analysis results

SRA elements / Assets	R	TS	RT	TST	R Upd.	TS Upd.
Confidentiality of information	8	2	5	5*	3	0
Integrity of information	8	7	5	5*	3	1
Availability of information	11	13	1	2	0	0
Accountability of information	4	6	0	0	0	0
Common	0	15	0	0	0	5
Total	31	43	11	7	6	6

to be adjusted, with respect to likelihood values, based on the security testing results.

5 Conclusion

The combinations of security risk analysis and security testing must be clearly distinguished depending on whether it is viewed from a security testing perspective (RST), or a security risk analysis perspective (TSR). Additionally, the immediate challenges that need to be addressed in these approaches are: (1) the gap between high-level security risk analysis results and low-level security test cases, and (2) the correct reflection of investable effort. An industrial case study evaluation of a TSR based approach showed how security testing can be used as a significant means to validate and update the risk model – i.e. approximately 20% of all risks and 14% of all threat scenarios had to be adjusted, with respect to likelihood values, based on the security testing results.

References

1. The Committee on National Security Systems: National Information Assurance (IA) Glossary, CNSS Instruction No. 4009, 2010.
2. ISO/IEC 27005:2011(E): Information technology - Security techniques - Information security risk management.
3. Erdogan, G., Meland, P.H., Mathieson, D.: Security Testing in Agile Web Application Development - A Case Study Using the EAST Methodology. In Proceedings of XP'2010. pp.14–27, 2010.
4. Solheim, I., Stølen. K.: Technology research explained. SINTEF Report, A313. Technical report, SINTEF Information and Communication Technology, 2007.
5. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis: The CORAS Approach. Springer, 2011.
6. British Computer Society Specialist Interest Group in Software Testing: BS 7925-2 Software testing. Software component testing, 1998.
7. Working Group 26 (WG26) of the ISO/IEC JTC1/SC7 Software and Systems Engineering committee. `http://www.softwaretestingstandard.org/` Last date accessed 2012-02-07.
8. International Organization for Standardization: ISO 31000 Risk management - Principles and guidelines, 2009.