Security Testing Approaches – For Research, Industry and Standardization

Axel Rennoch(✉), Ina Schieferdecker, and Jürgen Großmann

Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
{axel.rennoch, ina.schieferdecker,
juergen.grossmann}@fokus.fraunhofer.de

Abstract. Recently, in the Security testing domain a lot of knowledge has been collected from a significant amount of research. The contribution provides an introduction to advanced security testing methods and techniques in the context of European research and standardization projects. This includes numerous guidelines and best practices that have been identified and are applied in the context of industrial case studies. In particular it addresses risk modeling, security test pattern, functional security tests as well as fuzz testing, as important contributions to systematic, automatized test approaches in research, industry and standardization.

Keywords: Model-based security testing · Risk analysis · Test automation · Fuzzing

1 Introduction

In our daily life people all over the world rely on systems and services provided via open communication interfaces. Today many of these access points are used to exchange information that need to be protected since that protect critical infrastructures or are private.

As pointed out by the Software Engineering Institute, US, 2009: "The security of a software-intensive system is directly related to the quality of its software". In particular, over 90 % of software security incidents are caused by attackers exploiting known software defects. DIAMONDS [2] addresses this increasing need for systematic security testing methods by developing techniques and tools that can efficiently be used to secure networked applications in different domains. The RASEN project [9] is a follow-up project that especially addresses security assessments of large scale networked systems through the combination of security risk assessment and security testing, taking into account the context in which the system is used, such as liability, legal and organizational issues as well as technical issues.

2 The DIAMONDS and RASEN Projects

In the DIAMONDS project 23 partners from six European countries have been worked together over two and a half years to build up an innovative approach for model-based security testing applicable in different industrial domains.

Y. Yuan et al. (Eds.): ISCTCS 2013, CCIS 426, pp. 397–406, 2014.
DOI: 10.1007/978-3-662-43908-1_49, © Springer-Verlag Berlin Heidelberg 2014

DIAMONDS introduced four main innovations in the field of security testing methods and technologies:

- Advanced model-based security testing methods that combine different techniques to obtain improved results applicable to multi-domain security.
- Development of autonomous testing techniques based on automatic monitoring techniques to improve resilience of dynamically evolving systems.
- Pre-standardization work on multi-domain security test methodologies and test patterns to offer interoperable security test techniques and tools.
- Open source platform for security test tool integration to provide a common platform, which provides the user a single user interface towards various test tools, as well as a single reporting interface to have concise report from the various tools.

The RASEN project is a follow-up project with partners from DIAMONDS and deals with the subject of risk-based security testing, especially with:

- Compositional approaches that allows for conducting risk assessments for smaller parts or aspects of a system and systematically compose these assessments to obtain a global risk picture.
- Techniques, tools and methods to derive security test cases from security risk assessment results and to verify and update of the security risk assessment based on security test results.
- Continuous security assessment in which the security assessment is performed iteratively in such a way that results from previous assessments can be reused, and the security risk assessment picture can be rapidly updated with respect to changes in the system and its environment.
- Assess the risks related to non-compliance to legal norms by developing methods for risk assessments, which specifically take into account legal aspects of relevance to security.

Both projects aim at building a pre-standard for model-based security testing targeting heterogeneous and distributed systems and services and represent the enabling technology necessary for the introduction of formal security testing in industry.

3 Methods and Tools

3.1 Test Process

From the testing perspective one of the major changes is the integration of risk identification, prioritization, test selection and result consideration for the used security models.

Risk-based security testing approaches help to optimize the overall test process. The result of the risk assessment, i.e. the identified vulnerabilities, threat scenarios and unwanted incidents, are used to guide the test identification and may complement requirements engineering results with systematic information concerning the threats and vulnerabilities of a system. A comprehensive risk assessment additionally introduces the notion of probabilities and consequences related to threat scenarios. These risk values can be additionally used to weight threat scenarios and thus help identifying which threat scenarios are more relevant and thus identifying the ones that need to be treated and tested more carefully. We have identified the following two extensions to traditional testing or security testing processes.

- Risk-based security test planning: The goal of risk-based security test planning is to systematically improve the testing process during the test-planning phase. Risk assessment is used to identify high-risk areas or features of the system under test (SUT) and thus determine and optimize the respective test effort that is needed to verify the related security functionality or address related vulnerabilities. Moreover, selected test strategies and techniques are identified that are dedicated the most critical sources of risk (e.g. potential vulnerabilities and threat scenarios).
- Risk-based security test selection: Finding an optimal set of security test cases requires an appropriate selection strategy. Such a strategy takes the available test budget into account and also provides, as far as possible, the necessary test coverage. In functional testing, coverage is often described by the coverage of requirements or the coverage of model elements such as states, transitions or decisions. In risk-based testing coverage can be described in terms of the identified risks, their probabilities and consequences. Risk-based security test selection criteria can be used to control the selection or the selected generation of test cases. The criteria are designed by taking the risks as well as their probabilities and consequence values to set priorities for the test selections, test case generation as well as for the order of test execution.

Figure 1 is derived from ISO 29119 [8] and illustrates the extension of the traditional security testing process with risk assessment activities. The test planning (step 1) is the activity of developing the test plan. Depending on where in the project this process is implemented this may be a project test plan or a test plan for a specific phase, such as a system test plan, or a test plan for a specific type of testing (such as a performance test plan).

In step 2 risk assessment refers to the process of using risk assessment to support the test planning and test technique identification. The activity of test design and implementation (step 3) is the process of deriving the test cases and test procedures. Step 4 covers the test environment set-up and maintenance process and is the process of establishing and maintaining the environment in which tests are executed. In the context of step 5, risk assessment refers to the process of using risk assessment to prioritize the test cases which should be executed.

The test execution (step 6) is the process of running the test procedure resulting from the test design and implementation process on the test environment established by the test environment set-up and maintenance process. The test execution process may need to be performed a number of times as all the available test procedures may not be executed in a single iteration. This step is related to the activity of prioritizing and selecting tests to be executed. The selection criteria may e.g. be based on a risk assessment. The activity may also involve mutation/fuzzing of concrete executable test cases.

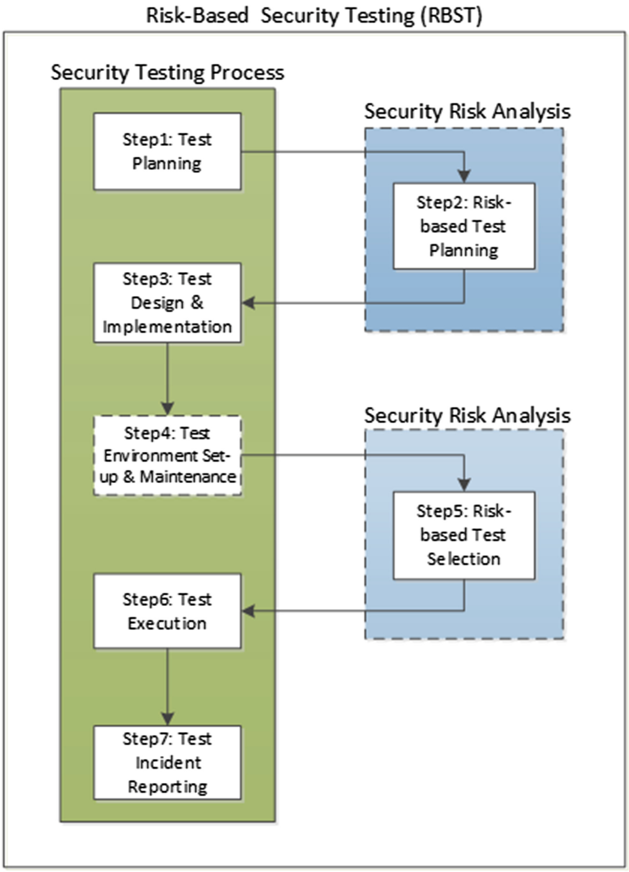
Risk-Based Security Testing (RBST)
Security Testing Process
Step1: Test Planning
Security Risk Analysis
Step2: Risk-based Test Planning
Step3: Test Design & Implementation
Step4: Test Environment Set-up & Maintenance
Security Risk Analysis
Step5: Risk-based Test Selection
Step6: Test Execution
Step7: Test Incident Reporting

Fig. 1. The generic process for risk-based security testing.

The test incident reporting (step 7) is the process of managing the test incidents. This process will be entered as a result of the identification of test failures, instances where something unusual or unexpected occurred during test execution, or when a retest passes. In test-based risk assessment, the incident reporting activity may involve an assessment of how the test results impact the risk picture and change the risk model.

3.2 Tools

Previous project results have shown that methods and tools used for model-based testing need to be combined, may depend on the industrial domains and have to be integrated in the productive environment of the customers (see e.g. D-MINT [5]). From this perspective it is not a surprise that we are facing a similar situation in the context of security testing.

Following the introduction of the generic model-based security process described above it is obvious that different procedure steps may require different tools. The DIAMONDS industrial case studies have used tools from both, the developments by project partners and existing tools from third parties. The following twelve tools have been created or extended by project partners during the project lifetime [3]. Even that some tools have multiple functions they may be assigned to three main different tool groups: modelling, active and passive testing.

Test Modelling and Generation Tools

- CORAS (Sintef) is a risk modelling tool with a graphical editor. It is based on a model-driven risk analysis method.
- FUZZINO (FhG Fokus): Fuzzino is a test data generator for fuzz testing. With fuzzing, it is possible to find security-related weaknesses in your code. It's about injecting invalid or unexpected input data to a system under test. That way, security-relevant vulnerabilities may be detected when malicious data is processed instead of rejected.
- CertifyIt (Smartesting): Model-based security testing from behavioral models and test purposes is an extension of functional model-based testing (MBT): The model for test generation captures the expected behaviour of the system under test (SUT). This model is dedicated for automated generation of security tests, and generally formalizes the security functions of the SUT but also the possible stimuli of an attacker as well as the expected answer of the SUT. The test purposes are test selection criteria that define the way to generate tests from the test generation model.
- RISKTest (FhG Fokus): is a tracing tool especially designed for security testing. It enables traceability between security testing artefacts, e.g. identified risk elements, system objects and test data. The traceability allows interaction and combination of different security engineering and testing tools and is the basis for determining coverage and completeness metrics like risk coverage.

Test Execution Tools

- Defensics (Codenomicon) uses behavioral models to alter the behavior to generate millions and millions of nearly-valid messages that systematically anomalies some parts of the information exchange to test the target system for robustness.
- TTworkbench (TestingTech) supports the entire lifecycle of TTCN-3 based tests with textual and graphical editors, a TTCN-3 to Java compiler, and a test execution management environment composed of graphical tracing, debugging and reporting facilities for centralized and distributed test components.
- KameleonFuzz (INT) automatically detects Type-1 and Type-2 Cross Site Scripting (XSS) in Web Application.
- TRICK tester is a platform used for penetration (intrusion) testing. It contains all the main software tools to test web applications and network systems like Web Servers.

Test Analysis and Monitoring Tools

- MMT (montimage) is a monitoring solution that combines: data capture; filtering and storage; events extraction and statistics collection; and, traffic analysis and reporting.
- LiSTT (INT) performs intra- and inter-procedural dataflow analysis on the binary code. The produced result is the set of vulnerable paths that have been detected with respect to an input source.
- TestSym-P (Telecom sudParis) aims at passively testing an Implementation under test (IUT) to verify if it respects the protocol requirements represented as IOSTS property.
- Malwasm (itrust) helps reverse engineers understand what a binary does, it can identify static and dynamic malware analysis.

Based on this toolset the following approach is proposed for the application of model-based security testing:

- Model, identify risks and navigate to learn about vulnerabilities in your system.
- Combine and (re)use selected DIAMONDS models, methods, techniques and tools to test and monitor countermeasures, verify security risks and discover vulnerabilities.
- Evaluate your models and risks and repeat the application of DIAMONDS model-based security testing approaches to enjoy a more secure system and life with your assets.

Figure 2 illustrate the co-operation of a set of four tools, two from DIAMONDS and two from other sources, which have been combined to build a model-based security testing platform: Risks models have been built with CORAS, ProR manages

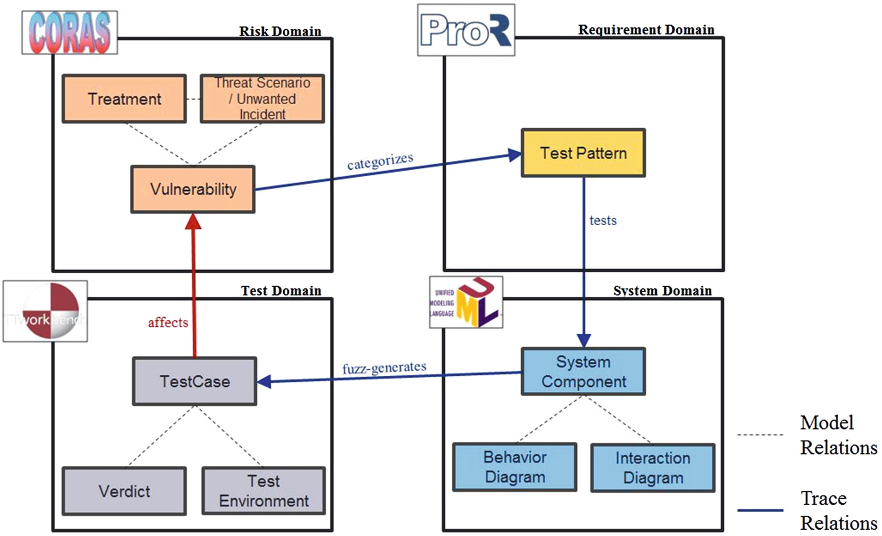

Fig. 2. Model-based security testing platform

the requirements, the UML tool introduces the use/test cases, and TTworkbench has been applied for test execution with TTCN-3 [6]. A detailed presentation of all DIAMONDS tools is available in [3].

4 Industrial Case Studies

The major focus of our work has been the industrial case studies in the following domains: Radio protocols, automotive, smart cards, telecommunication, banking and industrial automation.

One of the DIAMONDS case studies from the banking domain consists of a banknote processing system that counts and sorts banknotes depending on their currency, denomination, condition and authenticity. Banknote processing machines are used in central, large and medium banks and also in CITs (cash in transport) and other organizations that handle large amounts of banknotes. These machines are usually configured to work in a network as shown in Fig. 3. Currency processors, reconciliation stations, vault management systems and control centers are connected on a network either on a LAN or WAN.

Configuration and monitoring information is exchanged between the currency processor and the control center. The type of information exchanged requires a high degree of security.

The focus of security tests is on the currency processor and the reconciliation station. The currency processor as well as the reconciliation station was provided as virtual machines for VMware Workstation where external interfaces are replaced by simulation and were supplemented with snapshots. That allows creating a consistent state of the SUT before executing a test case and is necessary for batch execution of test cases. The test bed at Fraunhofer FOKUS consists of the two virtual machines, one for the currency processor and another for the reconciliation station. Windows 7-based host system runs the virtual machines. The main focus of security tests will be the components inside the virtual machines. The available interfaces are the Message Router (.Net Remoting implementation) over LAN, as well as keyboard, USB and other peripherals through the hardware abstraction layer of the virtual machine. There is a database running inside the virtual machine.

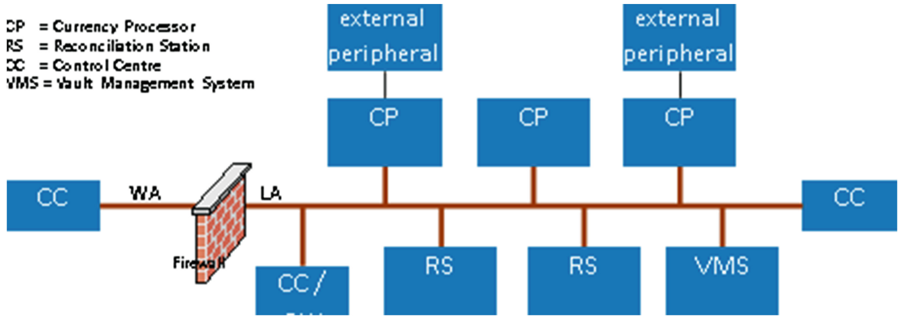

Fig. 3. Banknote counting machine.

The underlying threats were used as starting point for the risk analysis. A risk analysis following the CORAS approach [1] was performed and the potential vulnerabilities as well as the consequences of the threats were analyzed. CORAS provides several kinds of diagrams for different phases of the analysis. As a result of the risk analysis, several vulnerabilities were considered that should be tested whether they actually exists within the system under test (SUT). In order to generate appropriate tests for these vulnerabilities, security test patterns provide a suitable way to select test generation techniques or test procedures.

In order to test for the abovementioned vulnerabilities identified during risk analysis, both well established and new developed methods were applied to the system. Data fuzzing approaches for SQL injection were applied by a new developed fuzz testing extension for TTCN-3. Data fuzzing sends a large number of invalid values to the system under test at certain points within a test case.

Based on the risk models, 30 behavioral fuzz test cases were executed on the SUT regarding an authentication bypass. By executing these test cases, the risk of an authentication bypass using behavioral means was covered by applying behavioral fuzzing. The developed behavioral fuzzing approach extends existing functional test cases towards tests of security aspects. Therefore, the applied fuzzing approaches can take advantage of the effort made for functional testing of the SUT and do not require development of new test cases for security testing.

For further case studies reports please refer to ETSI TR 101 582 [7].

5 Standardization

One of the most important standardization bodies for IT security is the International Organization for Standardization (ISO). Its subcommittee SC27 is responsible for IT security techniques. This work also covers methods and techniques for security evaluation in the context of the common criteria (CCRA). The international ITU Telecommunication Standardization Sector concentrates on international standards for the telecom domain. Several documents have been identified in the context of security. Furthermore the European Telecommunication standardization institute (ETSI) has several technical committees (TC) and industrial specification groups (ISG) that are working for security techniques.

The review of the collected standardization activities results in the identification of ETSI as the first focus of DIAMONDS and RASEN standardization activities [4]. In a second step the results should be forwarded also to other international standardization bodies like e.g. ISO or ITU-T. The reason for selecting ETSI is mainly due to the memberships of DIAMONDS and RASEN partners in ETSI and the relative short timeframes for standardization work at ETSI. Three standardization groups have been identified for being the major technical group of the DIAMONDS and RASEN:

- ETSI TC MTS
- ETSI TC INT
- ETSI ISG ISI

The new ETSI TC MTS special interest group for security aspects has been working on security in three directions: modelling/specification (of system risks etc.), (paper-based) risk analysis, and testing that includes scanning (libs) on known attacks, functional/traditional testing, and negative testing to discover unknown vulnerabilities, configuration mistakes (using e.g. fuzzing, penetration). The work in TC INT has been concluded in the technical report: IMS/NGN Security Testing and Robustness Benchmark.

The work in ISG ISI will propose a way to produce security events and to test the effectiveness of existing detection means. Scenario catalogues will give inspiration for test/monitoring objectives and allow faster implementation. Examples of frequent security test pattern will be used to illustrate some powerful means and methods of event testing. The definition and use of a test (pattern) catalogue for test implementation allows: introduction of dedicated sets, reasonable efforts, and comparison of results.

Figure 4 illustrates our approach from project work package results to international standardization bodies.

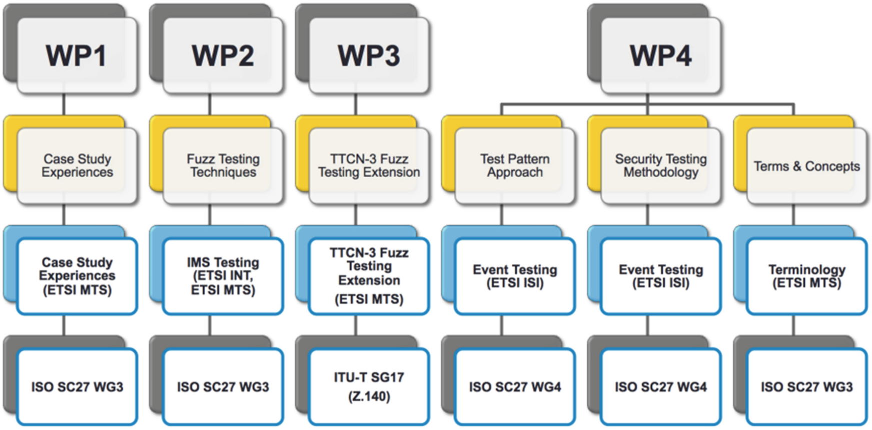

Fig. 4. Contributions from DIAMONDS work packages

6 Outlook

During the project lifetime several presentations and exhibition have been performed. The project approach and results have been accepted as a valuable contribution to the improvement to security testing. It has been proposed to initiate a community for advanced security testing tools. From the project viewpoint we like to remind security testing experts: Do IT with models.

We thank our colleagues from FOKUS and the other DIAMONDS and RASEN partners for the great work and cooperation in the project, as well as the ITEA and national authorities and the European Union's Seventh Framework Program (FP7/2007-2013) under grant agreement no 316853 for their support.

References

1. CORAS. Risk analysis method and tool. http://coras.sourceforge.net/
2. DIAMONDS project. http://www.itea2-diamonds.org
3. DIAMONDS deliverable D5.WP3: Final Security Testing Tools (May 2013)
4. DIAMONDS contributions to Standards Organizations. Project restricted deliverable (May 2013)
5. D-MINT project. http://www.fokus.fraunhofer.de/en/sqc/projekte/projekt_archiv/2009/D-MINT/index.html
6. ETSI ES 201 873: Testing and Test Control Notation, version 3 (TTCN-3), http://www.ttcn-3.org
7. ETSI TR 101 582: Methods for Testing and Specification (MTS); Security Testing; Case Study Experiences
8. International Standards Organization. ISO 29119 Software and system engineering - Software Testing-Part 2: Test process (draft) (2012)
9. RASEN project. http://www.rasenproject.eu/