Final draft ETSI EG 203 251 V1.1.1 (2015-11)

Methods for Testing & Specification;
Risk-based Security Assessment and Testing Methodologies

Reference
DEG/MTS-203251

Keywords
assurance, security, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights 4
Foreword 4
Modal verbs terminology 4
1 Scope 5
2 References 5
2.1 Normative references 5
2.2 Informative references 5
3 Definitions and abbreviations 6
3.1 Definitions 6
3.2 Abbreviations 7
4 Overview 8
5 Integration outline 9
5.1 Security risk assessment 9
5.2 Security testing 10
5.3 Combining the security testing and security risk assessment workstreams 10
5.4 System lifecycle integration 11
6 Test-based activities to security risk assessment 13
6.1 Integrating security testing in the security risk assessment workstream 13
6.2 Test-based security risk identification 14
6.3 Test-based security risk estimation 16
7 Risk-based activities to security testing 18
7.1 Integrating security risk assessment in the security testing workstream 18
7.2 Risk-based security test planning 19
7.3 Risk-based security test design and implementation 22
7.4 Risk-based test execution, analysis and summary 25
8 Managing complexity within system lifecycle 27
8.1 Composition and Decomposition 27
8.2 System Security Risk Assessment 28
8.3 Component Security Risk Assessment 28
8.4 Refinement and Update Process 29
8.5 Security Testing 29
Annex A: A conceptual model for risk-based security testing 30
A.1 Testing 30
A.2 Security Testing 30
A.3 Risk assessment 31
A.4 Security risk assessment 31
Annex B: Bibliography 33
History 34

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This final draft ETSI Guide (EG) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS), and is now submitted for the ETSI standards Membership Approval Procedure.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document describes a set of methodologies that combine security risk assessment and security testing activities in a systematic manner. This includes both risk assessment aimed to improve security testing and test based activities used to improve the security risk assessment. The methodologies are built upon a collection of consistently aligned activities with associated rules, methods and best practices. The activities are described in such a way that they provide guidance for the relevant actors in security testing and security risk assessment processes (i.e. actors in the role of a security tester, security test manager, and/or risk assessor). The activities and their level of specification are based on standards like ISO 31000 [i.10], IEEE™ 829-2008 [i.6] and ISO 29119 [i.9] so that they apply for a larger number of security testing and risk assessment processes on hand.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Alberts, Christopher & C., J. and Dorofee, Audrey. A. J.: "OCTAVE Threat Profiles". Software Engineering Institute, Carnegie Mellon University, Criteria Version 2.0, Technical report CMU/SEI-2001. http://www.cert.org/archive/pdf/OCTAVEthreatProfiles.pdf-TR-016. ESC-TR-2001-016, 2001.

[i.2] Broy M. and Stølen K.: "Specification and Development of Interactive Systems: Focus on Streams, Interfaces and Refinement". Springer, 2001.

[i.3] ETSI TS 102 165-1 (2011): "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis".

[i.4] Herzog, P.: OSSTMM 2.1. Open-Source Security Testing Methodology Manual; Institute for Security and Open Methodologies, 2003.

[i.5] Howard, M. & Leblanc, D. E.: "Writing Secure Code"; Microsoft Press, 2002.

[i.6] IEEE™ Standard for Software and System Test Documentation (IEEE™ 829-2008), ISBN 978-0-7381-5747-4, 2008.

[i.7] ISO 27000:2009(E): "Information technology -- Security techniques -- Information security management systems -- Overview and vocabulary", 2009.

[i.8] ISO/IEC/IEEE™ 29119: "Software and system engineering -- Software Testing -- Part 1: Concepts and definitions", 2012.

[i.9] ISO 29119: "Software and system engineering -- Software Testing -- Part 2: Test process", 2012.

[i.10] ISO 31000:2009(E): "Risk management -- Principles and guidelines", 2009.

[i.11] ISTQB Glossary of testing terms version 3.0.1.

NOTE: Available at http://www.istqb.org/downloads/finish/20/206.html, as of date 29.09.2015.

[i.12] James J. Cebula, L. R. Y.: "A Taxonomy of Operational Cyber Security Risks", Carnegie Mellon, Software Engineering Institute, CERT Program, 2010.

[i.13] Jones, Jack A.: "An Introduction to Factor Analysis of Information Risk (FAIR)".

NOTE: Available at http://www.riskmanagementinsight.com/media/docs/FAIR-introduction.pdf, as of date 29.09.2015.

[i.14] Masse, T.; O'Neil, S. & Rollins, J.: "The Department of Homeland Security's Risk Assessment Methodology: Evolution, Issues, and Options for Congress", The Department of Homeland Security's Risk Assessment Methodology, 2007.

[i.15] OMG: UML testing profile version 1.1 (formal/2012-04-01).

NOTE: Available at http://www.omg.org/spec/UTP/1.1, as of date 29.09.2015.

[i.16] Souza, E.; Gusmao, C. & Venancio, John Wack, Miles Tracy, M. S.: "Guideline on Network Security Testing -- Recommendations of the National Institute of Standards and Technology"; NIST Special Publication 800-42, 2003.

[i.17] Saitta, P.: Larcom, B. & Eddington, M.: Trike v.1 Methodology Document; 2005.

[i.18] Testing Standards Working Party. BS 7925-1: "Vocabulary of terms in software testing", 1998.

[i.19] Wing, J. M.: "A specifier's introduction to formal methods". IEEE™ Computer 23(9), 8, 10-22, 24, 1990.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

asset: anything that has value to stakeholders, its business operation and their continuity

consequence: outcome of an event affecting objectives [i.10]

event: occurrence or change of a particular set of circumstances [i.10]

likelihood: chance of something happening [i.10]

objective: something the stakeholder is aiming towards or a strategic position it is working to attain

risk: combination of the consequences of an event and the associated likelihood of occurrence (adapted from ISO 31000 [i.10]

risk level: magnitude of a risk or combination of risks, expressed in terms of the combination of consequences and their likelihood [i.10]

risk source: element which alone or in combination has the intrinsic potential to give rise to risk [i.10]

security requirement: specification of the required security for the system (adopted from [i.18])

security risk: risk caused by a threat exploiting a vulnerability and thereby violating a security requirement

security risk assessment: process of identifying, estimating and evaluating security risks

stakeholder: person or organization that can affect, be affected by, or perceive themselves to be affected by a decision or activity [i.10]

test case: set of preconditions, inputs (including actions, where applicable), and expected results, developed to determine whether or not the covered part of the *test item* has been implemented correctly

test completion criteria: set of generic and specific conditions, agreed upon with the stakeholders, for permitting a testing process or a testing sub process to be completed

test condition: testable aspect of the test item (i.e. a component or system), such as a function, transaction, feature, quality attribute, or structural element identified as a basis for testing

test coverage item: attribute or combination of attributes to be exercised by a *test case* that is derived from one or more test conditions by using a test design technique

test incident: event occurring during testing that requires investigation (adopted from ISTQB [i.11])

test incident report: detailed description for any unexpected incident or test that failed

test item: work product (e.g. system, software item, requirements document, design specification, user guide) that is an object of testing

test log: recording which tests cases were run, who ran them, in what order, and whether each test passed or failed

test plan: detailed description of test objectives to be achieved and the means and schedule for achieving them, organized to coordinate testing activities for some test item or set of test items

test procedure: sequence of *test cases* in execution order, and any associated actions that may be required to set up the initial preconditions and any wrap up activities post execution

test result: indication of whether or not a specific test case has passed or failed, i.e. if the actual result corresponds to the expected result or if deviations were observed [i.8]

test (design) technique: compilation of activities, concepts, processes, and patterns used to identify *test conditions* for a *test item*, derive corresponding test coverage items, and subsequently derive or select test cases

threat: potential cause of an unwanted incident [i.7]

vulnerability: weakness of an asset or control that can be exploited by a threat [i.7]

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CVSS	Common Vulnerability Scoring System
FAIR	Factor Analysis of Information Risk
ISO	International Organization for Standardization
ISTQB	International Software Testing Qualifications Board
OCTAVE	Operationally Critical Threat, Asset and Vulnerability Evaluation
SQL	Structured Query Language
SRA	Security Risk Analyst
SRAT	Security Risk Assessment Tool
ST	Security Tester
STET	Security Test Execution Tool
STMT	Security Test Management Tool
STST	Security Test Specification Tool
SUT	System Under Test
TM	security Test Manager
TVRA	Threat Vulnerability and Risk Analysis

UML	Unified Model Language
UTP	UML Testing Profile

4 Overview

The present document describes methodologies and their underlying activities that are dedicated to support companies and organizations in undertaking security assessments for large scale, networked systems. The methodologies cover security assessments on different level of abstraction and from different perspectives. Security risk assessment by itself can be applied with different goals in mind. Legal risk assessment especially addresses security threats in a legal context and under consideration of legal consequences. Security risk assessment specifically deals with the concise assessment of security threats, their estimated probabilities and their estimated consequences for a set of technical or business related assets. Finally, compliance assessment and security testing can be used to actually examine the target under assessment, i.e. an organization or system, for compliance issues or vulnerabilities.

Security testing is considered to discover flaws, vulnerabilities and other technical issues to security by applying test procedures to the actual system under test. In contrast, security risk assessment is meant to analyse potential threats to a system, often on a higher, non-technical level, by especially addressing legal or business related issues. The present document describes the systematic integration of security testing and security risk assessment. Integrating and interweaving the activities from both work streams, thus a systematic integration and completion of risk assessment activities with security testing results or the systematic guidance of security testing by means of risk assessment results, allows for a more precise, focused and dynamic assessment of the security of systems and associated processes.

In the following clauses the integration between security risk assessment and security testing is described in more detail. In clause 5 the overall integration approach is introduced. Clauses 6 and 7 precisely specify the aspects of integration. Clauses 5, 6 and 7 focus on a description on process level that is generic and that is applicable to all system lifecycle phases as well as to all kinds of security testing. Clause 8 shows that application of the integration in the different phases of a system lifecycle. All integration related activities are documented in a similar manner using the template shown in table 1.

Table 1: Template for documenting process activities

Name	The name of the activity
Actors	The actors that are referred to in the activity
Tools	The tools that are involved in the activity
Precondition	The condition that needs to be fulfilled before the activity could be initiated successfully.
Result	Describes the desired results of the activity.
Scenario	The scenario that describes the individual actions taken by the actors
Data exchanged/ processed	The data that are exchanged during the integration use case **In:** *The data that go into the activity. Terms from the conceptual model are used to describe the data.* **Out:** *The data that are the outcome of the activity. Terms from the conceptual model are used to describe the data.*

The possible actors and tools that can be referred to are described as follows:

Actors:

- **Security Risk Analyst (SRA):** The person responsible for doing the security risk assessment.
- **Security Test Manager (TM):** The person responsible for doing the security test management.
- **Security Tester (ST):** The person responsible for doing the security testing.

Tools:

- **Security Risk Assessment Tool (SRAT):** The tool that supports the security risk assessment.
- **Security Test Management Tool (STMT):** The tool that supports the security test management.
- **Security Test Specification Tool (STST):** The tool that supports the security test specification.
- **Security Test Execution Tool (STET):** The tool that supports the execution of test procedures and test cases.

The methodologies and activities have been developed and evaluated in the RASEN research project (www.rasenproject.eu).

5 Integration outline

5.1 Security risk assessment

Security risk assessment is an iterative process that analyses the potential threats to a system in order analyse their impact and to estimate the likelihood of their occurrence. The risk assessment comprises the identification of assets, threats and vulnerabilities as well as the identification, specification and realization of risk treatments (i.e. security controls and other countermeasures). Risk itself is a metric that relates the frequency and/or likelihood of unwanted incidents to their impact.

From a process point of view risk assessment is considered as the overall process of risk identification, risk estimation and risk evaluation.

- Risk identification is a set of activities dedicated to finding, recognizing and describing risks. This involves identifying sources of risk, areas of impacts, events (including changes in circumstances), their causes and their potential consequences. Risk identification can involve historical data, theoretical analysis, informed and expert opinions, and stakeholders' needs. It typically comprises a threat analysis as well as a vulnerability analysis.

- Risk estimation is the process of determining the level of risk. This involves developing an understanding of the nature of a risk, its sources and its consequences.

- Risk evaluation is the process of comparing the results of risk estimation with risk criteria to determine whether the risk and/or its magnitude is acceptable or tolerable. Risk evaluation assists in the decision about risk treatment and on the most appropriate risk treatment strategies and methods.

Currently there is a larger number of security risk assessment methods like ETSI TVRA [i.3], CVSS [i.14], STRIDE/DREAD [i.5], OCTAVE [i.1], FAIR [i.13] and Trike [i.17], which provide dedicated guidance on how to identify the sources of risks, their causes and their potential consequences within different contexts and with different strategies. Their main purpose is to provide systematic guidance and the definition of a consistent and unambiguous vocabulary for risk identification and handling. Security risk assessment can be qualitative or quantitative as well as informal (check-list based) or formal (model-based). Qualitative risk assessment is based on qualitative risk and quantitative risk assessment is based on some quantities, numbers, or measurements. In model-based security risk assessment, the security risk assessment is conducted with a language for the documentation of assessment results and a clearly defined process for conducting the assessment. In this regard the Carnegie Mellon University's Computer Emergency Response Team provides a taxonomy on operational cyber security risks [i.12]. The taxonomy identifies sources of operational cyber security risks and separates them into four classes. It distinguishes between risks caused by actions of people, by systems and technology failures, by failed internal processes, or by external events. Each class is broken down into further subclasses, which are described by individual elements (e.g. "actions of people" is subdivided into "Inadvertent Actions", "Deliberate Actions" and "Inaction"). The Factor Analysis of Information Risk (FAIR) [i.13] provides an information security risk taxonomy, which is comprised of two main branches according to the FAIR's overall risk definition "Risk = Loss Event Occurrence and Probable Loss Magnitude". The OCTAVE method defines the main tasks during risk assessment with threats identification, security measures identifications, definition of business impacts, and the definition of security measures' costs and their standardized values. A step by step approach eases the estimations on the individual risk factors. It starts with the definition of asset-based threat profiles. In this phase the members of an organization identify important information assets, the threats to those assets and the security requirements of the assets. A second phase targets the identification of infrastructure vulnerabilities. Especially the information technology infrastructure is examined for weaknesses (technology vulnerabilities) that can lead to unauthorized action. The last phase is dedicated to the development of a security strategy. The information generated by the organizational and information infrastructure evaluations are carefully analysed to identify risks to the organization and to the organization's mission as well as to identify countermeasures.

5.2 Security testing

The term security testing or software security testing designates activities that check the security properties of software. While a number of approaches have long been around targeting specific attacks on systems (e.g. vulnerability scanners), more systematic security testing of systems with respect to specified policies or security properties are a relatively new approach that has started to be addressed since around the year 2000. In general the software security testing activities can be divided into functional security testing, robustness testing, performance testing and penetration testing. While functional security testing, robustness testing and performance testing are used to check the functionality, availability, and efficiency of the specified and carefully planned security functionalities and systems (e.g. firewalls, authentication and authorization subsystems, access control), penetration testing or security vulnerability testing directly addresses the identification and discovery of system vulnerabilities undiscovered until a given point in time and caused by security design flaws. These kind of tests analyse systems for any potential vulnerabilities that may result from poor or improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures. Penetration test objectives are to determine feasibility of an attack and the impact of a successful exploit.

5.3 Combining the security testing and security risk assessment workstreams

The overall process of a combined security assessment is derived from ISO 31000 [i.10] and slightly extended to highlight the identification and evaluation of compliance and quality issues as one of the major tasks that need to be carefully aligned with typical risk assessment activities. It is defined independent of any application domain and independent from the level, target or depth of the security assessment. It can be applied to legal risk and compliance assessment as well as for any kind of technical security assessment and testing processes.

Figure 1 shows the main activities of a combined risk assessment and security testing process. It starts with a preparatory phase called "*Establishing the context"* that includes preparatory activities like "*Understanding the Business and Regulatory Environment*" as well as the "*Requirements & Process Identification*". During the first phase the high level security objectives are identified and fixed. The latter phase is meant to analyse and document the technical context of the target under assessment. Moreover, the figure shows additional support activities like "*Communication & consult"* and "*Monitoring and review"* that are meant to set up the management perspective, thus to continuously control, react, and improve all relevant information and results of the process. From a process point of view these activities are meant to provide the contextual and management related information for the combined security assessment and are considered to be common for security risk assessment workstream as .well as for the test-based risk assessment workstream.

The main part, namely the "*Security Assessment"*, covers the integration between the risk assessment workstream and a security testing workstream. It consists of a combination of typical security risk assessment activities that are defined in ISO 31000 [i.10] and typical security testing activities that follow testing standards like ISO 29119 [i.9].

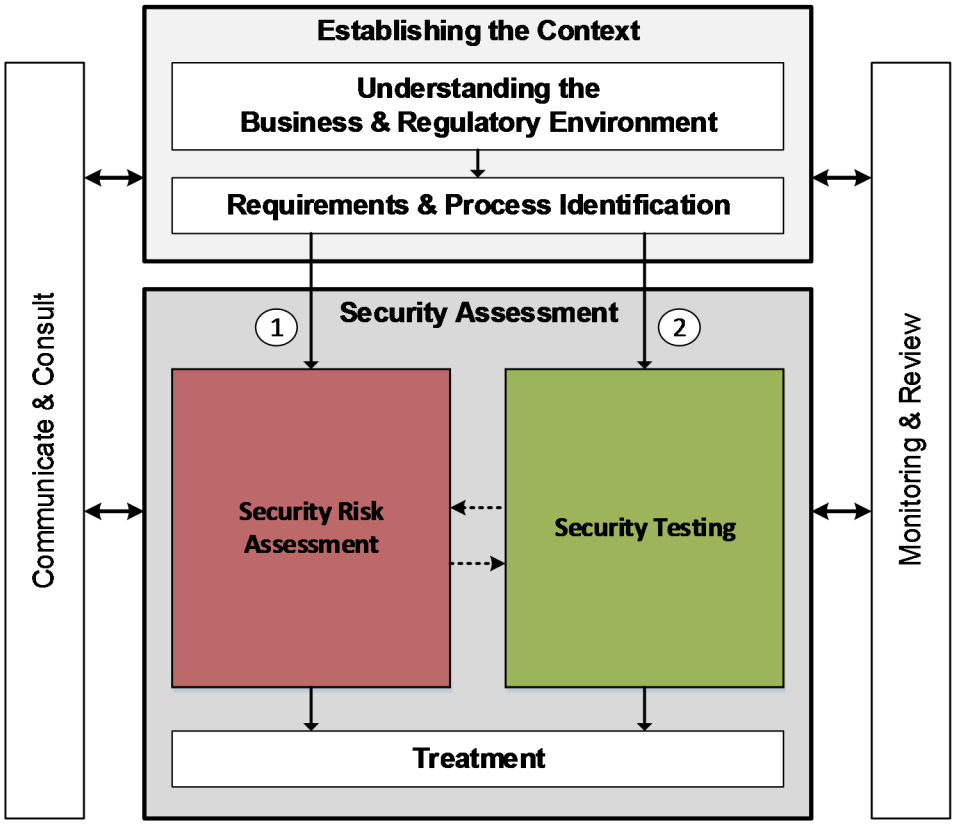

Figure 1: Main activities of a combined risk assessment and security testing process

The present document distinguishes two main perspectives, each represented by a set of activities that are combined to form a workstream carried out during system development or operation.

1) A test-based security risk assessment workstream should start like a typical risk assessment workstream and should use testing results to guide and improve the risk assessment. Security testing is used to provide feedback on actually existing vulnerabilities that have not been covered during risk assessment or allows to adjust risk values on basis of tangible measurements like test results. Security testing should provide a concise feedback whether the properties of the target under assessment have been really met by the risk analysis.

2) The risk-based security testing workstream should start like a typical testing workstream and uses risk assessment results to guide and focus the testing. Such a workstream should start with identifying the areas of risk within the target's business processes and building and prioritizing the testing program around these risks. In this setting risks help focusing the testing resources on the areas that are most likely to cause concern or supporting the selection of test techniques dedicated to already identified threat scenarios.

5.4 System lifecycle integration

As depicted in figure 2, risk-based security testing and test-based risk assessment can be applied in different phases and to different testing activities in the system lifecycle.

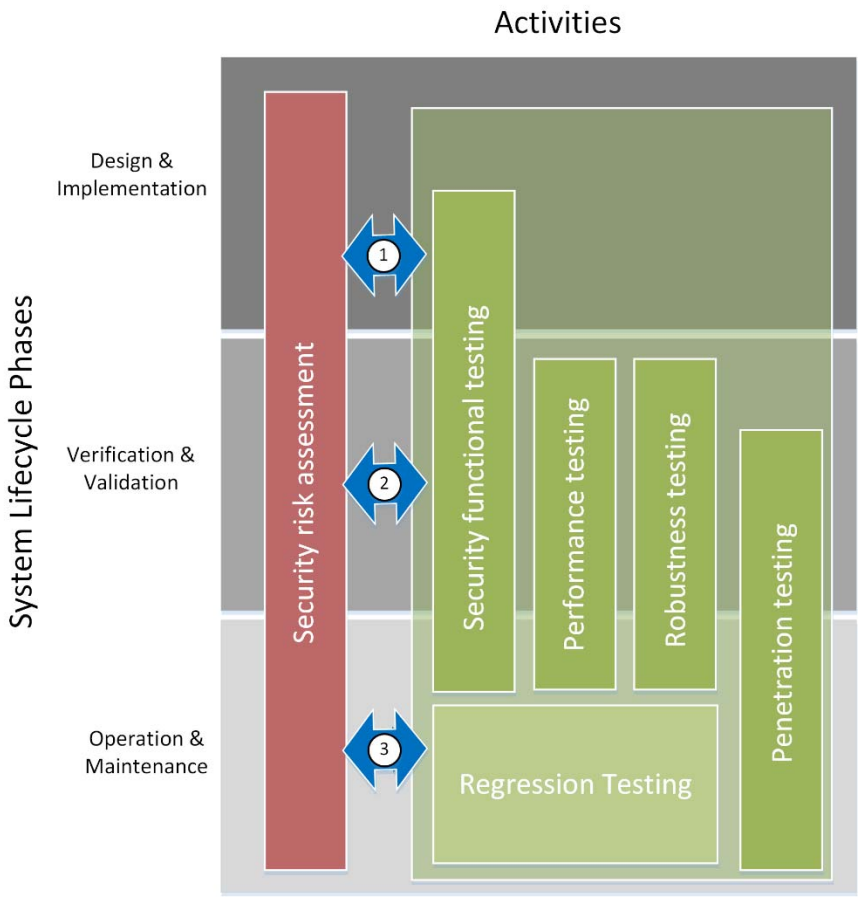

Figure 2: Risk-based security testing as systematic combination between security risk assessment and security testing

1) During design and implementation risk-based security testing and test-based risk assessment should focus the integration between security risk-assessment and security functional testing. The main points of reference are security functional requirements and the verification of their implementation by testing. The notion of risk might help to focus the implementation and testing efforts for all development driven testing activities (e.g. module and unit testing).

2) During the verification and validation phase security testing can (but not necessarily will) be extended to also cover the other security testing activities like performance testing, robustness testing and penetration testing. Risk-based security testing should be used to focus the test design and test implementation efforts, to choose the appropriate testing techniques and to just communicate or to relate the test results to risks.

3) During the operation and maintenance phase the focus of security testing slightly changes towards regression and penetration testing. Penetration testing is used to discover new and unknown vulnerabilities. The potential exploitation of these newly discovered vulnerabilities can constitute the new risks that should be integrated in the risk assessment. Regression testing is typically used to verify whether a changed system still meets the original security requirements with respect to functionality, performance and robustness.

6 Test-based activities to security risk assessment

6.1 Integrating security testing in the security risk assessment workstream

The main purpose of integrating the testing process into the risk assessment process is to use testing to extend some of the activities of the risk assessment process and thus to improve the overall process results. This is achieved by ensuring that test results are used as explicit input to the risk assessment. Figure 3 shows how the overall security assessment process (shown in figure 1) is refined into a process for test-based risk assessment. Here the risk assessment activity has been decomposed into the three activities Risk Identification, Risk Estimation and Risk Evaluation. These three, together with the "*Establishing the Context*" and "Treatment" activities form the core of the ISO 31000 risk management process. As indicated in figure 3, there are in particular two places where testing can in principle enhance the risk assessment process. The first, denoted 1 in the figure, is during risk identification. In a risk assessment process, the risk identification activity is performed with respect to a target of analysis which is described and documented in the "*Establishing the Context*" phase. In a test-based risk assessment setting however, the risk identification is not only based on the documentation of the target of analysis, but also on relevant test results of the target of analysis. Particularly relevant in this setting is testing using automated testing tools such as vulnerability scanners or network discovery tools.

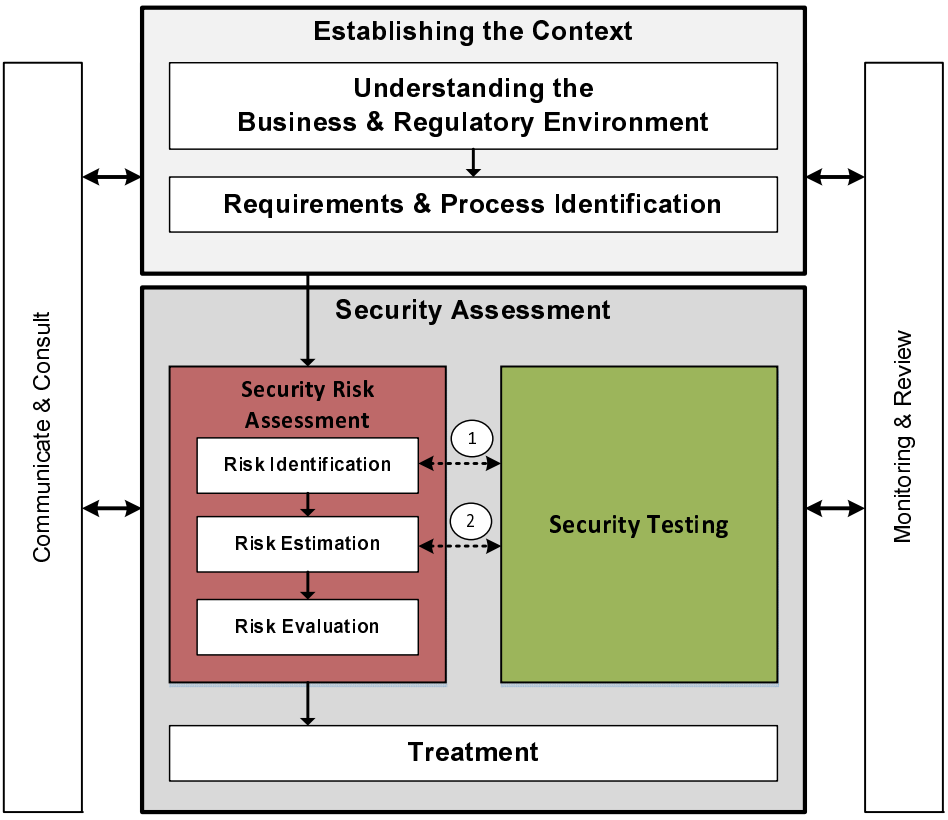

Figure 3: Generic workstream for test-based security risk assessment

The second risk assessment activity that can be enhanced by the testing process (denoted 2 in figure 3) is risk estimation. The main reason for doing testing here is to gain increased confidence in the correctness of the risk model. In particular, the likelihood estimates of the risk model might have a low confidence if they e.g. depend on vulnerabilities whose presence in the target of analysis is unknown. By doing testing in this setting, it is investigated whether such vulnerabilities really are present in the target of analysis, and then use the test results to update the confidence level of the risk model.

6.2 Test-based security risk identification

Risk identification is the process of finding, recognizing and describing risks. This involves identifying sources of risk (e.g. threats and vulnerabilities), areas of impacts (e.g. the assets), events (including changes in circumstances), their causes and their potential consequences. It should disclose the analysis of the potential threat or attack surface, the identification of potential threat and vulnerabilities and the derivation of complete threat scenarios, covering the relations between threats, vulnerabilities and unwanted incidents. Risk identification can involve historical data, theoretical analysis, informed and expert opinions, and stakeholders' needs [i.10].

A test-based security risk identification improves security risk identification through information on the actual system. Security testing is able to identify/indicate actual vulnerabilities or areas of an actual system that are potentially vulnerable. This kind of testing may be performed by e.g. use of network discovering techniques or vulnerabilities scanners.

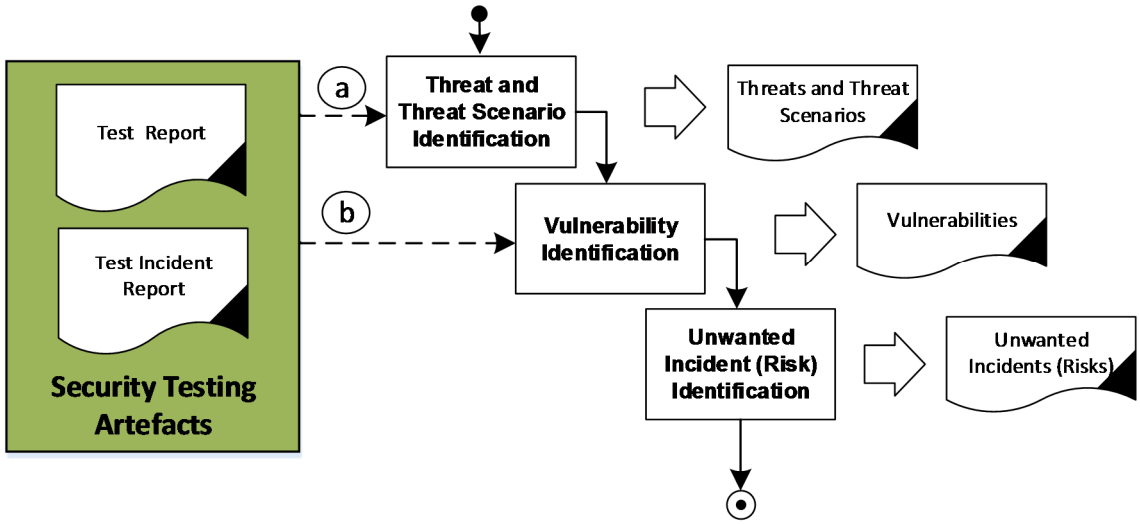

Figure 4: Test-based security risk identification

In figure 4, it is shown how the risk identification can be structured. As indicated in the figure, there are in particular two activities (see arrows a and b) that can be integrated with testing:

a) Test-based attack surface analysis

b) Test-based vulnerability identification

The purpose of the threat and threat scenario identification activity is to identify threats and threat scenarios. A threat may be human or non-human, malicious or non-malicious. A hacker is an example of a typical malicious human threat. A threat scenario is a series of events that is initiated by a threat and that may lead to an unwanted incident. A cyber security attack such as SQL injection is a typical example of a threat scenario. Testing can be used in order to obtain information that can support the identification of threats and threat scenarios. Particularly relevant in this setting are testing techniques that yield information about the interfaces/entry points, the attack-surface, and potential attacks against the target of evaluation. The kinds of testing tools that can be used for this purpose are network discovery tools, web-crawlers, static code analysis tools, and fuzz testing tools. Table 2 describes a test-based attack surface analysis as supporting activity during threat and threat scenario identification.

**Table 2: Test-based security risk identification:
Test-based attack surface analysis (a)**

Name	**Test-based attack surface analysis (a)**
Actors	Security Risk Analyst (SRA)
Tools	Security Risk Assessment Tool (SRAT), Security Test Execution Tool (STET)
Precondition	A test report covering results from attack surface analysis. A model or other specification documents describing the interfaces of the target of assessment.
Result	A detailed definition of the attack surface of the target of assessment.
Scenario	1) The SRA should analyse the system model, other specification document, and publicly available information to identify the attack surface of the target of assessment. 2) The SRA should initiate a semi-automatic or automatic scan of the system/network to detect hidden entry points for attacks. The results are documented by means of a scan or test report. 3) Based on this analysis, the SRA should indicate which features or areas of the target of assessment should be prioritized in the risk identification step.
Data exchanged/ processed	**In:** *Specification documents, test report* **Out:** *Attack surface definition (with prioritization of areas/features)*

NOTE: For a semi-automatic or automatic scan of the target of assessment, the following tool categories should be considered:

- Static analysis tools that check the code
- Vulnerability scanners or network discovery tools
- Crawlers, spiders or other dynamic web site discovery tools
- Fuzz-testing tools

Test-based vulnerability identification refers to the use of testing to obtain information that supports the vulnerability identification activity. Testing techniques that yield information about the presence of actual vulnerabilities in the target of evaluation or potential vulnerabilities that may be present in the target of evaluation are relevant in this activity. The kind of testing tools that can be used for this purpose are penetration testing tools, static and dynamic code analysis tools, and vulnerability scanners. Table 3 describes a test-based threat and vulnerability identification supporting activity during threat and threat scenario identification.

**Table 3: Test-based security risk identification:
Test-based threat and vulnerability identification (b)**

Name	**Test-based threat and vulnerability identification (b)**
Actors	Security Risk Analyst (SRA)
Tools	Security Risk Assessment Tool (SRAT), Security Test Execution Tool (STET)
Precondition	A test incident report covering results from testing or vulnerability assessments.
Result	A detailed list of potential threats, potential vulnerabilities and actual vulnerabilities of the target of assessment.
Scenario	1) The SRA should analyse the system model, other specification document, and publicly available information to identify potential threats, potential vulnerabilities and already known vulnerabilities for the target of assessment. 2) The SRA should initiate the active exploration (e.g. penetration testing or semi-automatic or automatic scan) of the actual target of assessment to identify vulnerabilities or indicators for vulnerabilities. The results are documented by means of a test incident report showing the discovered vulnerabilities or indications thereof. 3) Based on this analysis, the SRA should indicate which vulnerabilities (and associated threats) of the target of assessment should be additionally handled in the risk identification step.
Data exchanged/ processed	**In:** *Specification documents, test report* **Out:** *Threat and vulnerability list*

NOTE: For a semi-automatic or automatic scan of the target of assessment, the following tool categories should be considered:

- Static analysis tools that check the code

- Vulnerability scanners or network discovery tools
- Fuzz-testing tools

6.3 Test-based security risk estimation

Accurate risk estimation is essential for a successful outcome of a risk assessment. However, risk estimation is one of the hardest activities of a risk assessment since the information basis for the estimation is often imprecise and insufficient, and analysts are often forced to rely on expert judgment. This might result in a high degree of uncertainty related to the correctness of the estimates.

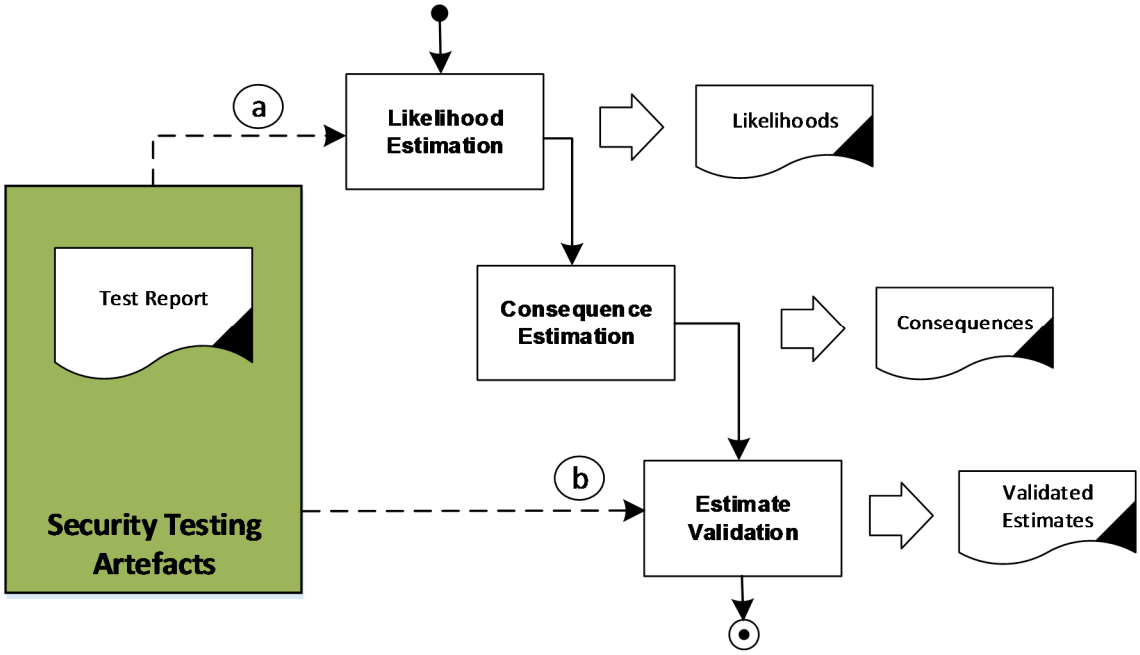

Figure 5: Test-based security risk evaluation

In this context, testing should be used to produce an additional input that allows for a precise characterization of some of the properties of a risk model. In particular, the likelihood estimates of the risk model might have a low confidence if they, e.g. depend on vulnerabilities whose presence in the target of analysis is unknown. By doing testing in this setting, it is investigated whether such vulnerabilities really are present in the target of analysis. Finally, the test results should be used to update the confidence level of the risk model.

As shown in figure 5, the risk estimation activity can be decomposed into the three sub-activities: Likelihood Estimation, Consequence Estimation, and Estimate Validation. The last sub-activity refers to checking and/or gaining confidence in the correctness of the risk estimates. As indicated in figure 5, there are in particular two activities that can be integrated with testing:

a) Test-based likelihood estimation

b) Test-based estimate validation

Likelihood estimation is the activity of estimating likelihoods for risks and their causes. In a security setting, this involves estimating the likelihood that: security attacks will be initiated; attacks will be successful if initiated; successful attacks will lead to identified risks. Likelihoods should be documented using the likelihood scales defined in the *Establishing the Context* step of the overall risk assessment process.

Testing is particularly relevant for obtaining information which can support the estimation of the likelihood that an attack will be successful if initiated. This is because security testing is most often used for identifying vulnerabilities, and the presence of these has a direct impact on this likelihood. Thus the testing techniques used for test-based likelihood estimation are similar to those used for test-based vulnerability identification (as described in clause 6.1). The main difference between these activities is that in the former, information about the vulnerabilities is only used as a means of supporting likelihood estimation (see table 4).

Table 4: Test-based security risk estimation: Test-based likelihood estimation (a)

Name	**Test-based likelihood estimation (a)**
Actors	Security Risk Analyst (ST)
Tools	Security Risk Assessment Tool (SRAT)
Precondition	A risk evaluation matrix, risk model with identified risks.
Result	A revised risk evaluation matrix showing the estimated likelihood values of the risks.
Scenario	1) The SRA should analyse the risk model and should identify elements that can be better understood when tested. 2) The SRA should initiate the testing (if not already initiated by other activities) of these elements and should receive the test report and the test incident report. 3) The ST links the items of the test report and test incident report to elements of the risk model and estimates the likelihood for individual risk model elements based on the information obtained through the testing.
Data exchanged/ processed	**In:** *Risk evaluation matrix, risk model, test report, test incident report* **Out:** *Risk model, risk evaluation matrix*

Validation is the activity of checking or gaining confidence in the correctness of the estimated risk values. In a test-based setting, it is recommended that uncertainty related to the correctness of an estimate is explicitly expressed. For instance, instead of using single likelihood values such as frequency or probability, intervals of likelihoods should be used to express the belief that the correct likelihood lies somewhere within the interval without knowing precisely where. Uncertainty can then be measured in terms of the breadth of the interval - the broader the intervals, the more uncertainty there is.

As for the likelihood estimation activity, testing is particularly useful for obtaining information that support the estimation of likelihood of successful attacks. The main difference between test-based likelihood estimation and test-based likelihood validation, is that in the former activity, testing is used to obtain the likelihood in the first place, whereas in the second activity, the purpose is to validate or gain confidence in the correctness of a likelihood value which has already been estimated. If uncertainty is expressed explicitly, the test results may be used to lower this uncertainty value. For instance if likelihood intervals are used, the test results may result in a diminution of the intervals. Recalculating the likelihood values of risks as a result of the updated uncertainty is a good way of showing how the test results have impacted the risks. The overall scenario for a test-based estimate validation is shown in table 5.

Table 5: Test-based security risk evaluation: Test-based estimate validation (b)

Name	**Test-based estimate validation (b)**
Actors	Security Risk Analyst (ST)
Tools	Security Risk Assessment Tool (SRAT)
Precondition	A risk evaluation matrix and risk model with identified risks and estimations for likelihood and consequences should be available.
Result	A list of risk categories or groups that allow for a better evaluation of the risks.
Scenario	1) The SRA should analyse the risk model and should identify elements that can be better understood when tested. 2) The SRA should initiate the testing of these elements (if not already initiated by other activities) and should receive the test report. 3) The ST links the items of the test report and test incident report to elements of the risk model and updates/revises the estimates for likelihood values of individual risk model elements based on the information obtained through the testing.
Data exchanged/ processed	**In:** *Risk evaluation matrix, Risk model, Test report, Test Incident Report* **Out:** *Risk model, Risk evaluation matrix*

7 Risk-based activities to security testing

7.1 Integrating security risk assessment in the security testing workstream

Risk-based activities to security testing help to optimize the overall testing process. The result of the risk assessment, i.e. the identified vulnerabilities and threat scenarios 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. Almost all the approaches that combine testing and risk assessment aid the testing by means of one of the following activities:

a) Risk-based resource, effort, test or feature **prioritization**: This activity supports testing by using risk assessment artefacts to prioritize efforts and artefacts during test planning, test design, test implementation, test execution and/or test summary.

b) Risk-based test or test technique **identification**: This activity supports testing by using risk assessment artefacts (typically from fault/threat/vulnerability modelling) to identify test purposes, test techniques and test condition.

c) Risk based test **scenario derivation**: This activity supports testing by using risk assessment artefacts (together with a test model) to manually derive or automatically generate test scenarios or test cases.

General technical recommendations on security testing techniques [i.4], [i.16] propose the use of risk analysis results to guide security testing. These recommendations are very general in nature and describe in this sense no real method for risk-based testing.

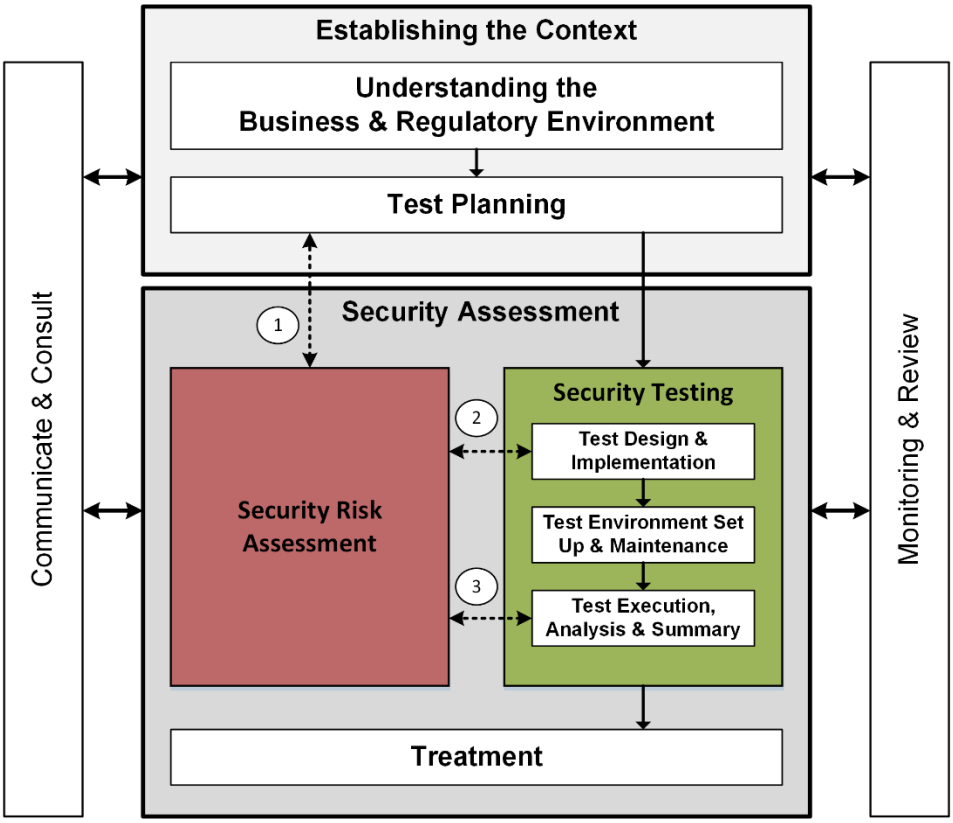

Figure 6: Process model for risk-based security testing

From a process point of view, the interaction between risk assessment and testing could best be described following the phases of a typical testing process. Figure 6 illustrates the three phases of a testing process that are affected and supported by risk-based security testing (denoted by dotted arrows 1, 2 and 3 in figure 6). In the following, the present document describes these phases and the related activities in more detail:

1) **Risk-based security test planning** deals with the integration of security risk assessment in the test planning process. Hence, security risk assessment is used to roughly 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 to address the related vulnerabilities. Moreover, a first assessment of the identified vulnerabilities and threat scenarios may help to select test strategies and techniques that are dedicated to deal with the most critical security risks.

2) **Risk-based security test design, implementation** deals with the integration of security risk assessment in the test design, implementation and execution process. During the test design and implementation phase, test cases are derived, implemented and assembled to test procedures. Security-risk assessment in general provides two different kinds of information that are useful within this process. On the one hand it provides detailed information on expected threats and potential vulnerabilities. This information can be used to systematically determine and identify test conditions (testable aspects of a system), test purposes or high-level test scenarios that are dedicated to address the identified threats and vulnerabilities. On the other hand, the security risk assessment provides quantitative estimations on the risk, i.e. the product of frequencies or probabilities and estimated consequences. This information can be used to select and prioritize either the test conditions or the actual tests when they are assembled to test set.

3) **Risk-based test execution, analysis and summary** deals with a risk-based test execution as well as with the systematic analysis and summary of test results. The decision on how extensive testing should be is always a question of the remaining test budget, the remaining time and the probability to discover further critical errors, vulnerabilities or design flaws. Risk-based test execution allows the prioritization of already existing test cases, test sets or test procedure during regression testing. Risk-based security test analysis and summary aims at improving the evaluation of the test progress by introducing the notion of risk coverage and remaining risks based on the intermediate test results as well as on the errors, vulnerabilities or flaws that have been found at a point in time. This process supports the test management process with risk related information that can be used to depict the test results in terms of their relation to the overall security risks.

While security test planning as well as security test execution, analysis and summary are more closely related to the test management process than security test design and implementation, all processes belong to the dynamic test process that is controlled by the test management process.

7.2 Risk-based security test planning

The test planning is the activity of developing the test plan. According to ISO 29119 [i.9] it determines the test objective, the test scope, and the risks associated to the overall testing process. The main outcome of this activity is the test strategy to be used and a plan that depicts the staffing, the required resources and a schedule for the individual testing activities. Figure 7 shows the integration of security risk assessment results in the overall test planning process. In the following, three integration activities have been outlined that all serve different purposes:

a) Integrate risk analysis

b) Risk-based test strategy design

c) Risk-based security resource planning and test scheduling

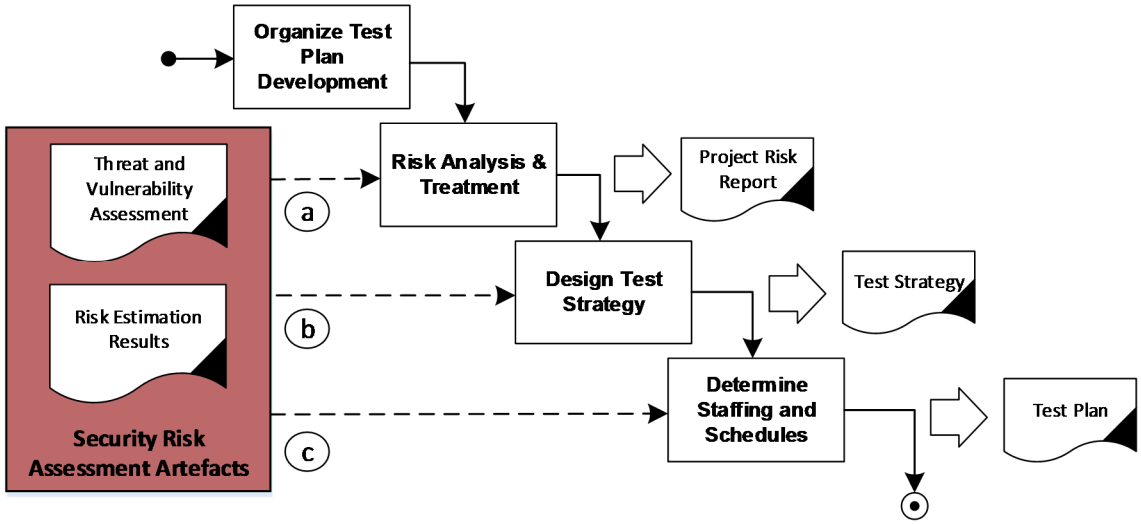

Figure 7: Process model for risk-based security test planning

Typically, risk analysis is a substantial part of the test planning process. The risk analysis is done to get an estimate on the specific project risks, considering the availability of test resources, specific product risks and other project related issues. The security risk assessment typically addresses the security risk of the product (i.e. the test item). As such, this kind of risk assessment can serve the project risk assessment with valuable estimates on the major product risks.

Table 6: Risk-based security test planning: Integrate risk analysis (a)

Name	**Integrated risk analysis (a)**
Actors	Security Test Manager (TM), Security risk analyst (SRA)
Tools	Risk Assessment Tool (SRAT), Security Test Management Tool (STMT)
Precondition	a) Contextual information like legal or regulatory requirements, organizational test and security policies, organizational or higher-level test strategies, and technical limitations as well as resource limitations are known. b) Security risk assessment results (threat, vulnerability and risk estimations) that capture the technical, business, regulatory and legal requirements are available.
Result	A project risk assessment that provides an overall risk picture for the test project, considering project risk that reflect risks that come from the security risk analysis.
Scenario	1) The Test Manager should review the relevant security risks to identify those, which have a special role for security testing. 2) The Test Manager should try to identify additional risks like other product risks or project related risks like missing resources, technical issues related to the test infrastructure, etc. 3) The Test Manager should develop an overall risk picture for the test project and communicate the risk picture to the Stakeholders.
Artefacts exchanged/ processed	**In:** *Vulnerabilities, threat scenarios, unwanted incidents, likelihoods, consequences, risk level* **Out:** *project risks*

One of the major activities during test planning is the design of a test strategy. A test strategy defines the test phases, the types of testing, the test techniques and the test completion criteria. For security testing especially the identification of test techniques is a challenge that should be optimized by directly considering the potential threats and vulnerabilities, which have been identified during a security risk identification.

Table 7: Risk-based security test planning: Risk-based security test strategy design (b)

Name	**Risk-based security test strategy design (b)**
Actors	Security Test Manager (TM), Security Risk Analyst (SRA)
Tools	Risk Assessment Tool (SRAT), Security Test Management Tool (STMT)
Precondition	a) Contextual information like legal or regulatory requirements, organizational test and security policies, organizational or higher-level test strategies, and technical limitations as well as resource limitations are known. b) Security risk assessment results (threat, vulnerability and risk estimations) that capture the technical, business, regulatory and legal requirements are available. c) Security risks that are relevant for testing have been identified, see integrated risk analysis (a).
Result	A test strategy comprising test phases, test types, features to be tested, test techniques and test completion criteria that directly address the identified threats and vulnerabilities.
Scenario	1) The Test Manager should assign vulnerabilities and threat scenarios to test items (interfaces, operations, components) and/or test conditions. 2) The Test Manager should try to identify the potential vulnerabilities that have the highest impact on the overall security risks when they are detected. 3) The Test Manager should assign test techniques that are capable to detect the identified vulnerabilities to each test item and/or to each test condition. 4) The Test Manager should assign test completion criteria to each test item and/or to each test condition. 5) The Test Manager should prioritize test items and/or test conditions by considering the required test efforts to match the completion criteria and the impact testing may have on the overall security risks (i.e. when vulnerabilities are detected or test suites pass without detecting anything).
Artefacts exchanged/ processed	**In:** *Vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level* **Out:** *List of applicable test techniques, test completion criteria, prioritized list of test items and/or test conditions*

When defining a security test strategy, the following sources of information should be considered:

- Rules and regulations that apply to the test item or the processes related to the test item
- Policies, objectives, and the strategies that are in place at the organization
- Publicly available security best practices (e.g. test pattern libraries and attack pattern libraries)
- Publicly available vulnerability scores (e.g. detectability, occurrence and impact scores)

The second major activity during test planning is the planning of resources and the schedule for the testing activities. Since the main task of security testing is finding vulnerabilities, resource planning and test schedules should be aligned with the major security risks so that resources and the order of testing allow for a focused testing of the test items or test condition where the detection of vulnerabilities shows the largest impact.

**Table 8: Risk-based security test planning:
Risk-based security resource planning and test scheduling (c)**

Name	**Risk-based security resource planning and test scheduling (c)**
Actors	Security Test Manager (TM)
Tools	Risk Assessment Tool (SRAT), Security Test Management Tool (STMT)
Precondition	a) Contextual information like legal or regulatory requirements, organizational test and security policies, organizational or higher-level test strategy, technical and resource limitation are known. b) Security risk assessment results (threat, vulnerability and risk estimations) are available that capture the technical, business, regulatory and legal requirements. c) Test strategy depicting the test items, test conditions, test techniques, etc.
Result	A test plan that depicts resources, staffing and test schedules respecting certain threats and vulnerabilities and their associated risk scores.
Scenario	1) The Test Manager should check for required security testing competences and should acquire new competences if certain security testing task require these competences. Security risk assessment results may indicate these competences (e.g. when certain potential vulnerabilities or threats need to be addressed). 2) The Test Manager should allocate resources considering the required test efforts for the test items or test conditions where testing may have the largest impact in terms of treating or minimizing the identified security risks. 3) The Test Manager should plan the test schedules in a way that test items or test conditions where testing might have the largest impact in terms of treating or minimizing the identified security risks are tested first.
Artefacts exchanged/ processed	**In:** *Vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level* **Out:** *Resource allocation and test schedules that respect the identified security risks*

In summary, the integration of security testing and security risk assessment is addressed during the test planning phase by three activities, that each contribute with the notion of security risks, threat scenarios and vulnerabilities to the testing activities.

7.3 Risk-based security test design and implementation

The test design and implementation process is mainly dedicated to derive the test cases and test procedures that are later on applied to the system under test. To achieve this in a systematic way the overall process should start with a concise definition of the features and test conditions that are the main subjects to test. based on that, the relevant test coverage items should be identified, the test cases should be derived and they finally should be assembled to adequate test sets and test procedures. Considering especially security testing, security risks, potential threat scenarios and potential vulnerabilities provide a good guidance on which of the features and test conditions require testing, which coverage items should be covered in what depth and how individual test cases and test procedures should look like.

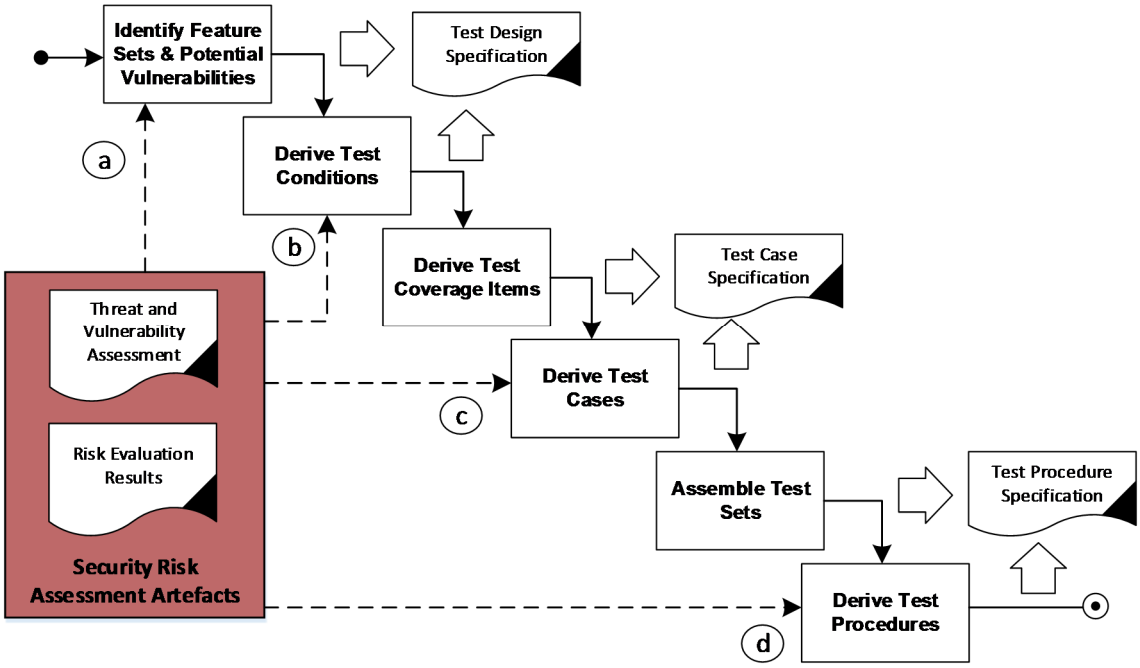

Figure 8: Process model for risk-based security test design

A first step during the test design phase is the identification and categorization of the security features that should be tested. Since security features describe functional security measures, this approach especially allows for testing the correctness of the feature implementation. Security risk assessment can be used to determine the most critical security features so that these features are tested more intensively and in more detail.

**Table 9: Risk-based security test design:
Risk-based identification and prioritization of feature sets (a)**

Name	**Risk-based identification and prioritization of feature sets (a)**
Actors	Security Tester (ST), Security Risk Analyst (SRA)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Security features are documented and the security risk assessment is available.
Result	Security features to be tested are grouped with respect to potential vulnerabilities and threat scenarios.
Scenario	1) The Security Tester should identify testable security features that need to be covered by security testing. This is done by grouping security features to feature sets that each addresses threat scenarios and/or vulnerabilities that have been identified during security risk assessment. 2) The Security Tester should prioritize the security feature sets using the risk levels that are associated with the threat scenario/vulnerabilities. 3) The Security Tester should document the relations between security feature sets and their associated threat scenarios and/or vulnerabilities (maintain traceability).
Data exchanged/ processed	**In:** *Vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, Risk Level* **Out:** *Prioritized list of testable security features (security feature sets)*

After a set of testable security features have been identified the security tester should derive the test conditions and test coverage items. This could be done on the basis of the identified features (see Risk-based identification and prioritization of features sets (a)) but needs to consider that especially security is a non-functional property and that a correct implementation of all security features may not ensure a secure system. Thus, additional test conditions and coverage items that especially address the detection of currently unknown vulnerabilities (vulnerability and robustness testing) need to be derived. Security risk assessment should be used to provide guidance for the derivation of test conditions and test coverage items for vulnerability and robustness testing.

**Table 10: Risk-based security test design:
Risk-based derivation of test conditions and test coverage items (b)**

Name	**Risk-based derivation of test conditions and test coverage items (b)**
Actors	Security Tester (ST)
Tools	Security Tester (ST), Security Risk Analyst (SRA)
Precondition	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Result	Test conditions and test coverage items weighted according to the impact testing may have on the overall associated security risks.
Scenario	1) The security tester should identify test conditions on the basis of the security features, threat scenarios and/or vulnerabilities that have been identified during security risk assessment and/or during a **risk-based identification and prioritization of features sets (a)**. Please note: Testing security features is one approach to security testing that is often not sufficient to cover all major threat scenarios and vulnerabilities. Thus a Security Tester should check whether all relevant threat scenarios are already covered by **risk-based identification and prioritization of features sets (a)** or if there are remaining risks from potential threat scenarios and vulnerabilities exist that still need to be covered by adequate test conditions. 2) The Test Designer should identify test coverage items corresponding to the test conditions identified in 1). Test coverage items and the respective test depth should be chosen according to the impact testing may have on the overall associated security risks.
Data exchanged/ processed	**In:** *Security feature sets, vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level, testable sets of security features* **Out:** *Test conditions and test coverage items weighted according to the impact testing may have on the overall associated security risks*

In the next step, the security tester should derive test cases on basis of test conditions and test coverage items. The security tester determines the preconditions for the individual test, he selects adequate input values and the actions to apply the selected test coverage items, and determines the expected results. Since security risk assessment has been performed to identify the test conditions and the test coverage items, then it was already considered through the previous activities. However, threat scenarios and potential vulnerabilities that have been identified during risk assessment might still help towards the identification of the preconditions, input values, actions and expected results.

Table 11: Risk-based security test design: Threat scenario based derivation of test cases (c)

Name	**Threat scenario based derivation of test cases (c)**
Actors	Security Tester (ST)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Testable security features, test conditions and test coverage items are known.
Result	Security test cases that address threat scenarios and potential vulnerabilities.
Scenario	1) The Test Designer should identify the preconditions for the tests, the test data, the test actions and the expected results by examining the test conditions, test coverage items, threat scenarios and potential vulnerabilities. 2) The Security Tester should document the relations between test cases, security feature sets and threat scenarios and/or vulnerabilities (maintain traceability). 3) The Security Tester and a Security Risk Analyst should review the test case specification and their coverage of threat and potential vulnerabilities identified by the security risk assessment.
Data exchanged/ processed	**In:** *Test conditions, test coverage items, vulnerabilities, threat scenarios, unwanted incidents, likelihoods, consequences, risk level, testable sets of security features* **Out:** *Security test cases*

Finally, the test cases should be assembled to test sets and test procedures. While test sets group test cases with common constraints on test environment or test items, test procedures defines the order of test execution and thus have to respect the pre- and postconditions. Security risk assessment should be used to prioritize the order test cases and thus the order of testing with respect to the associated risks.

Table 12: Risk-based security test design: Risk-based assembly of test procedures (d)

Name	**Risk-based assembly of test procedures (d)**
Actors	Security Tester (ST)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	*Test cases are available and associated with threat scenarios and potential vulnerabilities.*
Result	*Test procedures that are ordered with respect to their relevance.*
Scenario	1) The Test Designer should assemble test sets and test procedures in such a way that the most relevant tests are executed first. The most relevant test cases are the test cases that address the most critical risks. 2) The Test Designer should assemble test sets and test procedures in such a way that the post- and precondition of the individual test cases match.
Data exchanged/ processed	**In:** *Test cases, vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level, testable sets of security features* **Out:** *Security test procedures*

7.4 Risk-based test execution, analysis and summary

The decision on how extensive testing should be is always a question of the remaining test budget, the remaining time and the probability to discover further critical errors, vulnerabilities or design flaws. Risk-based security test analysis and summary aims at improving the evaluation of the test progress by introducing the notion of risk coverage and remaining risks on basis of the intermediate test results as well as on basis of the errors, vulnerabilities or flaws that have been found until a given point in time. This process supports the test management process with risk related information that can be used to depict the test results in terms of their relation to the overall security risks. In the following, three integration activities are outlined, namely:

a) Risk-based test execution prioritization

b) Risk-based test log analysis

c) Risk-based test summary creation

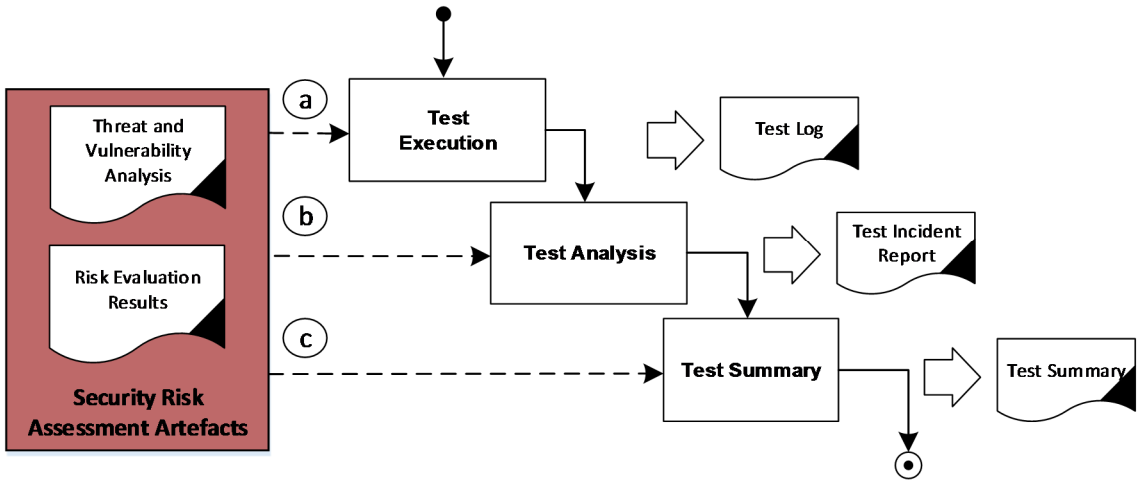

Figure 9: Process model for risk-based test execution, analysis and summary

Normally the execution order for test cases and test procedures is determined at test design by the assembly of test procedures. However, there are a number of regression test scenarios where reprioritization becomes necessary. In this case a risk-based approach for test executions prioritization may help to cover the most relevant remaining security risks.

Table 13: Risk-based test execution, analysis and summary: Risked-based test execution prioritization (a)

Name	**Risked-based test execution prioritization (a)**
Actors	Security Tester (ST)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Test cases and/or test procedures are available and associated with threat scenarios and potential vulnerabilities. The test environment is configured and ready to run the tests.
Result	Test execution that respects the criticality of addressed threats, vulnerabilities and/or features.
Scenario	1) The ST should prioritize test cases and test procedures in such a way that the most relevant tests are executed first. The most relevant test cases are the test cases that address the most critical risks. 2) The ST should run the test cases and/or test procedures.
Data exchanged/ processed	**In:** *Test cases, test procedures, risk level* **Out:** *Test logs*

The test analysis process is used for the evaluation of the test results and the reporting of test incidents. This process will be entered after the test execution and it mainly covers the analysis and evaluation of test failures and issues where something unusual or unexpected occurred during test execution. Its main purpose is to categorize the issues that occurred during testing and put them into context so that they can be rated by the test manager.

Table 14: Risk-based test execution, analysis and summary: Risked-based test result analysis (b)

Name	**Risked-based test result analysis (a)**
Actors	Security Tester (ST)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Test cases have been executed.
Result	New and/or updated incident are reported and assigned to either already detected vulnerabilities or to new vulnerabilities. Incidents that probably constitute new actual vulnerabilities are communicated so that they could be considered in the security risk assessment and/or the development.
Scenario	1) The Security Tester should analyse the test results (e.g., the test logs) and identify new incidents. 2) The Security Tester should classify newly identified incidents by means of their relation to artefacts from the security risk assessment (e.g., risks, threat scenarios, vulnerabilities). 3) The Security Tester should prioritize the newly identified incidents by means of associated artefacts from the security risk assessment. Issues related to critical risks should be rated higher than the ones that are associated with minor risks. 4) New and/or updated incidents are communicated to the relevant stakeholders.
Data exchanged/ processed	**In:** *Test logs, security risk assessment artefacts (vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level)* **Out:** *Incident report*

Finally, the overall test results, i.e. the test verdicts, the issues and their categorization are summarized in a way, that the stakeholder could understand the outcome of the tests.

Table 15: Risk-based test execution, analysis and summary: Risked-based test summary creation (b)

Name	**Risked-based test summary creation (b)**
Actors	Security Tester (ST)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Test cases have been executed. Test cases already have a traceable relation to security risk assessment artefacts.
Result	The test results are summarized respecting their relation to the a-priori identified security risks. The test report contains coverage of security risks.
Scenario	1) The Security Tester should analyse the test logs and separate security risks that have been tested successfully (all tests are passed) and those that have not been tested successfully (issues have been found). 2) The Security Tester should (re-) characterize the security risks by interpreting the test results. Therefore, the security tester should make use of dedicate test metrics to determine the quality of test procedures and thus the significance and validity of the test results.
Data exchanged/ processed	**In:** *Test logs, security risk assessment artefacts (vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level)* **Out:** *Test summary*

8 Managing complexity within system lifecycle

8.1 Composition and Decomposition

This clause provides guidance in applying the security assessment principles from clauses 5, 6 and 7 to a typical system lifecycle where decomposition and composition principles play a major role. In such a setting, the security assessment process itself should be compositional.

Composition and decomposition are well known principles of managing complexity in software engineering and system development [i.19]. Decomposition is the process of partitioning a system specification into separate modules that can be developed and analysed independently, thus breaking the development problem into more manageable pieces. Moreover, each module may be developed at different sites, by independent teams, or within different companies [i.2]. Composition is the opposite process. The term refers to the systematic integration of parts to realize the overall system or a system of systems.

A compositional process to security assessment should initially follow the same procedure as the (non-compositional) security assessment process. With respect to that, the system is decomposed into components or parts and each of these components are assessed individually. This has several advantages. It allows to consider specific contextual and technical details that become only visible when a system is broken down into several functional parts. Moreover, it supports processes with large integration efforts where multiple software or component supplier deliver individual parts of a system. For each of these components there can be a separate risk assessment that will be integrated to form the overall system's view.

Figure 10 illustrates the application of decomposition and composition in a typical software development lifecycle, where the target of analysis is assessed as a whole at the beginning and in parts or components, when the target is decomposed into several parts or components. Risk assessment, security testing and the integration thereof follow in principal the same decomposition/composition strategy as the target of assessment itself.

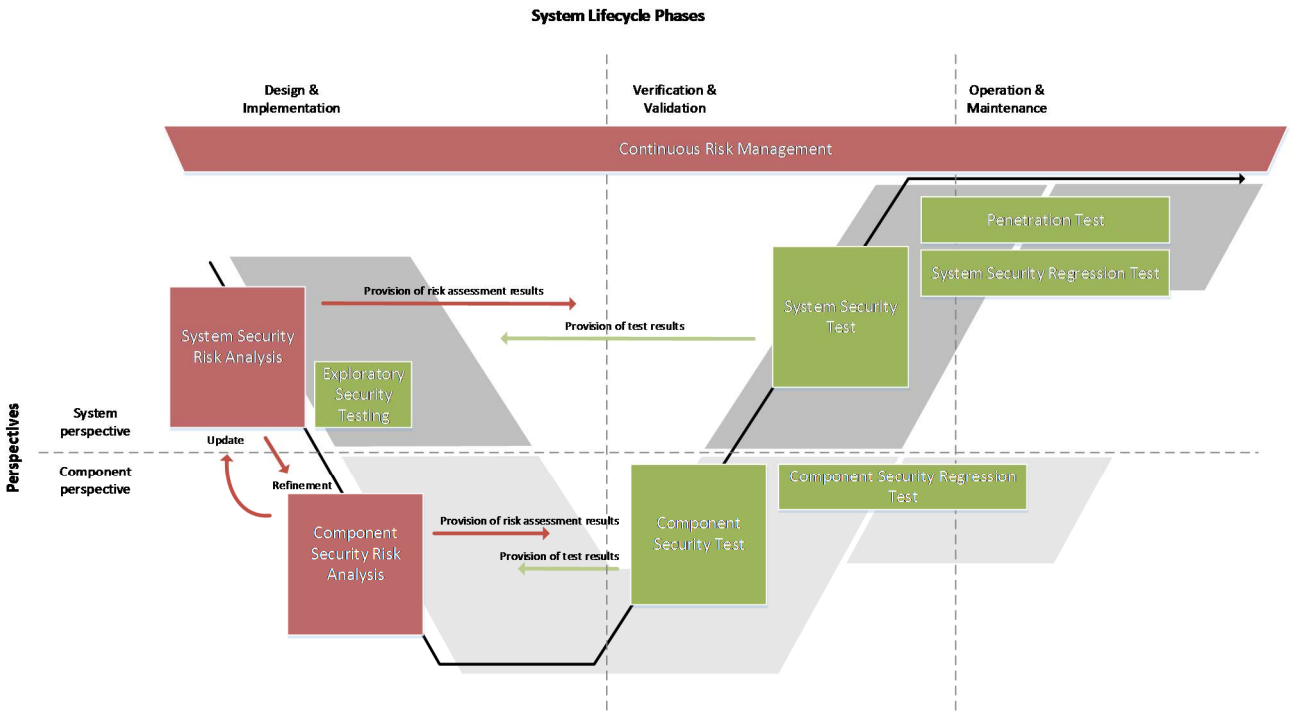

Figure 10: Overview of a risk assessment process with composition/decomposition

In order to achieve security through the overall lifecycle, the artefacts representing the outcomes of the security risk assessment phases and the testing phases should be managed and stored systematically so that consistency with and traceability to artefacts from earlier lifecycle stages is guaranteed.

8.2 System Security Risk Assessment

The overall process should start by taking the system's perspective. The system security risk assessment should start by following the risk assessment workstream that is described in clause 6. After having established the context, the SRA should go through the risk identification, risk estimation and risk evaluation phases. The risk assessment targets risks for the whole system, thus system related assets and incidents are considered. The definition of these assets and incident are dependent on the knowledge of the system's operational context. They could be technical as well as business related; the later only, if the business context is known. The interaction with security testing should in general follow the rules defined in clauses 6.2 and 6.3. If there is no established security testing process at that time, there should be a dedicated exploratory testing phase, which is driven by the risk assessment and only meant to provide dedicated testing feedback to the risk assessment. However, at an early stage in a system development process, there is often neither an established security testing process nor an existing system. In this case, the feedback from the security testing should be postponed until there is a functional system.

8.3 Component Security Risk Assessment

After having completed the risk assessment for the overall system, the system is typically decomposed into parts. In principle, the decomposition is driven by the development process and respects modularization requirements that come from the system's architecture or that are determined by integrator/supplier relationships. In fact, each of the components that have been defined during system development should be assessed on their own. However, clustering of components is allowed and might help to focus efforts on the major architectural items. In contrast to system risk assessment, the direct assets and incidents that are focused during component security risk assessment are mostly of technical nature. Thus, vulnerability assessment and the assessment of the technical impacts should get much more attention than threat and asset identification. Threat and asset identification is usually done on system level and should be deliberately reused during component security risk assessment. In principle, the component security risk assessment should be carried out, having already the component security testing phase in mind. Thus, assessment results and reports should be structured in such a way that they serve as input for the security testing workstream that is defined in clause 7.

8.4 Refinement and Update Process

The two processes, the system security risk assessment and the component security risk assessment, belong together. The relation between the two should be seen as an iterative refinement and update process. Security risk assessment provides the overall context. It identifies the high level assets (e.g. often determined by the business context of the system) and defines the overall threats, threat scenarios, vulnerabilities and unwanted incidents. The component security risk analysis allows for a deeper understanding of the technical causes and impacts focussing on vulnerabilities and unwanted incidents. Since component risk analysis is carried out at a later point in time, there is much more system related information available (e.g. interface definitions, details of realization). This information can be used to allow for a better localization and specification of vulnerabilities, unwanted incident and their impact on and propagation to other parts and components of the system. Finally, component security risk assessment results should be used to update the system security risk assessment with respect to estimates on probabilities, identified vulnerabilities and technical impact.

8.5 Security Testing

Security testing should start when security risk assessment has already gone through its first iteration. Thus, first risk assessment results are available for the system's perspective as well as for the component's perspective. Security test planning should be done according to clause 7.2 and cover both perspectives, i.e. the security system testing as well as security component testing phase. Security component testing should be used to test for vulnerabilities and the correctness of security features on component level. System security testing should be used to test the integrated system, cover integration & configuration related vulnerabilities and ensure (as far as testing alone can ensure) the functional correctness of the high level security features. The interaction with security risk assessment should in general follow the rules defined in clauses 7.3 and 7.4. While system security testing should especially interact with system security risk assessment, security component testing should interact with component security risk assessment.

Similar to system security risk assessment, the interaction between component security risk assessment and security testing should in general follow the rules defined in clauses 6.2 and 6.3. In contrast to security risk assessment.

Please note, especially when it comes to component level testing, static testing activities like source code analysis should be used in addition to dynamic testing. Static testing activities have a quite good discovery rate for a larger number of known vulnerabilities.

The overall security risk assessment and testing process that is described above should be considered as a highly iterative process. System security risk assessment should be used to guide and focus the component security risk assessment activities as well as the system security testing activities. In return, the component security risk assessment as well as the system security testing should be used to provide updates for the system security risk assessment. Similar, component security risk assessment should be used to directly improve the component security testing activities. In return, the results from component security testing should be used to update the component security risk assessment and thus, transitively, the system security risk assessment.

Operation & maintenance should be considered to be a "mini-lifecycle", potentially reflecting all preceding stages. Test planning, test design and test summary and execution should keep their relation to security risk assessment as described above and in clauses 7.1 to 7.4. An overall risk management workstream should ensure, that the risk assessment on the different level and the integration of the testing activities are kept up to date and in sync.

Annex A:
A conceptual model for risk-based security testing

A.1 Testing

Standards like IEEE™ 829 [i.6], ISO/IEC/IEEE™ 29119 [i.8], the ISTQB Glossary of testing terms [i.11], and the UML Testing Profile (UTP) [i.15] define the basic activities and related artefacts of a testing process. The major activities can be characterized as follows:

- Test planning (results: a test plan containing test conditions, test techniques, test coverage items and test completion criteria)
- Test design & implementation (results: test cases and test procedures)
- Test execution (results: test logs and test results)
- Test evaluation & incident reporting (result: test incidents reports and test incidents)

Since the present document focuses on the relationship between risk assessment and testing, the following model especially reflects the terms and concepts that are relevant to describe the interfaces between security testing and security risk assessment. In this sense the model concentrates on activities like test planning and test specification as well as the management, evaluation and interpretation of the test results. The following model is mainly based on terms and concepts taken from ISO/IEC/IEEE™ 29119 [i.8].

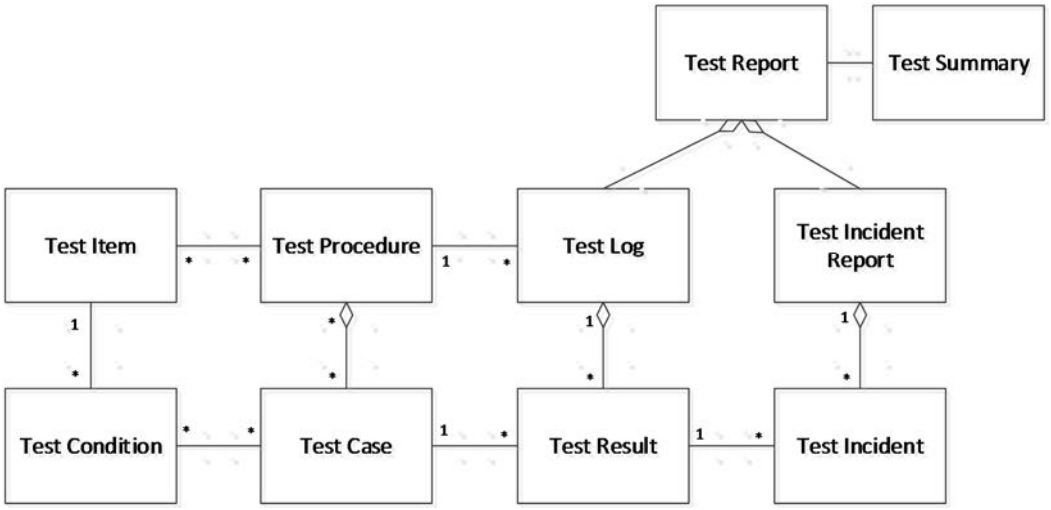

Figure A.1: Basic testing concepts

A.2 Security Testing

Security testing is used to experimentally check software implementations with respect to their security properties and their resistance to attacks. Functional security testing checks if the software security functions are implemented correctly and consistent with the security functional requirements. It is used to check the functionality, efficiency and availability of the specified security features of a test item. Security vulnerability testing directly addresses the identification and discovery of yet undiscovered system vulnerabilities. This kind of security testing targets the identification of design and implementation faults that lead to vulnerabilities that may harm the availability, confidentiality and integrity of the test item.

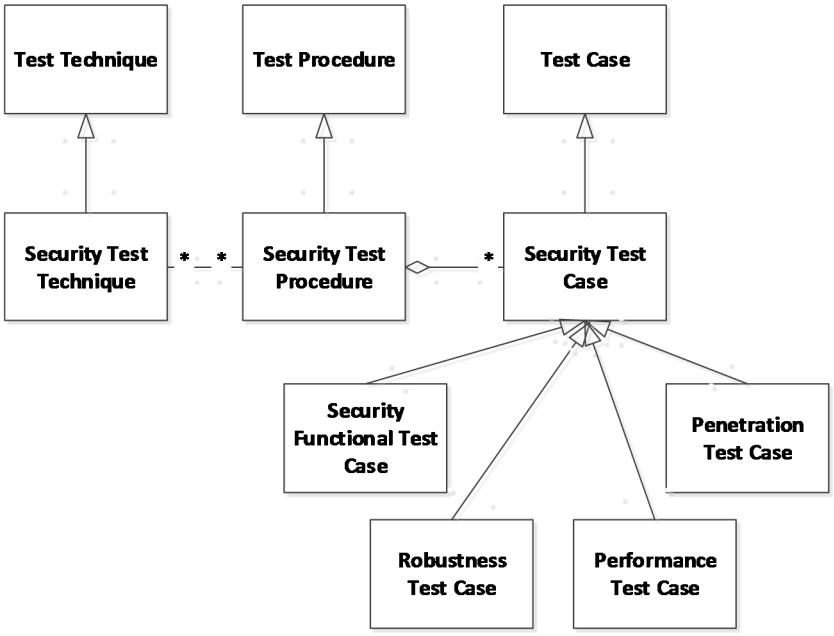

Figure A.2: Security testing

A.3 Risk assessment

The conceptual model and notions defined here are based on the ISO 31000 standard [i.10].

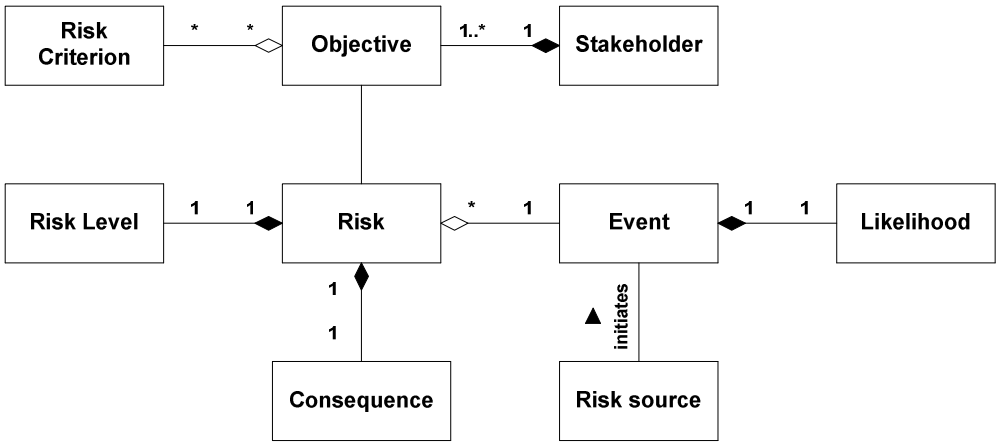

Figure A.3: Conceptual model for risk assessment

A.4 Security risk assessment

Lund et al. [i.11] classify risk analysis approaches into two main categories:

- Offensive approaches: Risk analysis concerned with balancing potential gain against risk of investment loss. This kind of risk analysis is more relevant within finance and political strategy making.
- Defensive approaches: Risk analysis concerned with protecting what is already there.

In the context of security, the defensive approach is the one that is relevant.

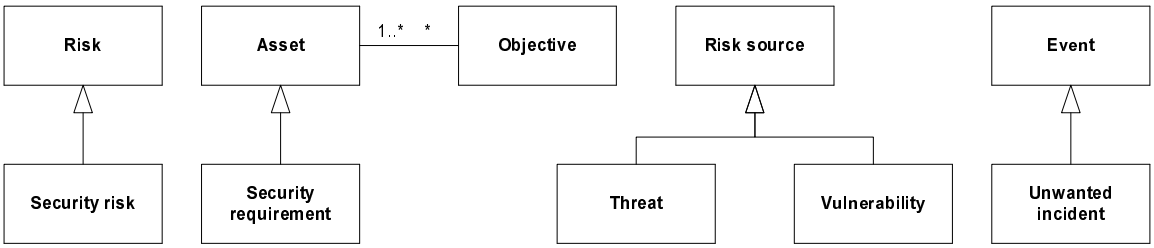

Figure A.4: Conceptual model for security risk assessment

The main terms related to security risk assessment and their relationship to previously defined terms in the risk assessment domain are illustrated in [i.7].

Annex B: Bibliography

Australian Standard AS 3806-2006 Compliance programs.

Amland, Risk-based testing: "Risk analysis fundamentals and metrics for software testing including a financial application case study". Journal of Systems and Software 53(3): 287-295 (2000).

Brændeland, G; Refsdal, A.; Stølen, K.: "Modular analysis and modelling of risk scenarios with dependencies". Journal of Systems and Software 83(10), 1995-2013 (2010).

IEEE™ Standard Glossary of Software Engineering Terminology (IEEE™ 610.12-1990), ISBN 1-55937-067-7, 1990.

ISO/IEC 15408: "Information technology -- Security techniques -- Evaluation criteria for IT security", 2008+.

ISO/IEC 27034: "Information technology -- Security techniques -- Application Security", 2011+.

ISO/IEC 30111: "Information technology -- Security techniques -- Vulnerability handling processes", 2013.

Mass Soldal Lund, Bjørnar Solhaug, Ketil Stølen: Model-Driven Risk Analysis, The CORAS Approach, Springer Verlag Berlin Heidelberg 2011, ISBN: 978-3-642-12322-1.

Masson A., M.-L. Potet, J.Julliand, R.Tissot, G.Debois, B.Legeard, B. Chetali, F. Bouquet, E. Jaffuel, L. Van Aertrick, J. Andronick, A. Haddad: An access control model based testing approach for smart card applications: Results of the POSE project, JIAS, Journal of Information Assurance and Security, 5(1), 335-351 (2010).

Michael, C. C. & Radosevich, W.: Risk-Based and Functional Security Testing; Cigital, Inc., 2005.

RASEN research project, RASEN Deliverable D5.3.1, Methodologies for Legal Compositional and Continuous Risk Assessment and Security Testing (v1), http://www.rasenproject.eu/deliverables, 2013.

RASEN research project, RASEN Deliverable D5.3.2, Methodologies for Legal Compositional and Continuous Risk Assessment and Security Testing (v2), http://www.rasenproject.eu/deliverables, 2014.

RASEN research project, RASEN Deliverable D5.3.3, Methodologies for Legal Compositional and Continuous Risk Assessment and Security Testing (v3), http://www.rasenproject.eu/deliverables, 2015.

Redmill, F. 2004. Exploring risk-based testing and its implications: Research Articles. Softw. Test. Verif. Reliab. 14, 1 (Mar. 2004), 3-15.

Redmill, F. 2005. Theory and practice of risk-based testing: Research Articles. Softw. Test. Verif. Reliab. 15, 1 (Mar. 2005), 3-20.

Testing Standards Working Party. BS 7925-1: "Vocabulary of terms in software testing", 1998.

Zech, P.: "Risk-Based Security Testing in Cloud Computing Environments"; PhD Symposium at the Fourth IEEE™ International Conference on Software Testing, Verification and Validation (ICST), 2011 Trust Management (IFIPTM'2009), pages 215-233, Springer, 2009.

Zimmermann, F.; Eschbach, R.; Kloos, J. & Bauer, T.: Risk-based Statistical Testing: A Refinement-based Approach to the Reliability Analysis of Safety-Critical Systems EWDC 2009: Proceedings of 12th European Workshop on Dependable Computing, HAL - CCSD, 2009.

History

Document history		
V1.1.1	November 2015	Membership Approval Procedure MV 20160122: 2015-11-23 to 2016-01-22