Certified Tester
Security Test Analyst Syllabus

Version 1.0

International Software Testing Qualifications Board



Copyright Notice

Copyright Notice © International Software Testing Qualifications Board (hereinafter called ISTQB®)

ISTQB® is a registered trademark of the International Software Testing Qualifications Board.

Copyright © 2026. Dr. Frank Simon (chair), Beata Karpinska, Maria Kispal, Ralf Pichler, Frans Dijkman, Chris-Julien Becker.

The Task Force wishes to thank Ian Howles (GWG) and Samuel Ouko (EWG) for their assistance and help in reviewing and creating his syllabus.

All rights reserved. The authors hereby transfer the copyright to the ISTQB®.

The authors (as current copyright holders) and ISTQB® (as the future copyright holder) have agreed to the following conditions of use:

Extracts, for non-commercial use, from this document may be copied if the source is acknowledged. Any Accredited Training Provider may use this syllabus as the basis for a training course if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus and provided that any advertisement of such a training course may mention the syllabus only after official Accreditation of the training materials has been received from an ISTQB®-recognized Member Board.

Any individual or group of individuals may use this syllabus as the basis for articles and books, if the authors and the ISTQB® are acknowledged as the source and copyright owners of the syllabus.

Any other use of this syllabus is prohibited without first obtaining the approval in writing of the ISTQB®.

Any ISTQB®-recognized Member Board may translate this syllabus provided they reproduce the abovementioned Copyright Notice in the translated version of the syllabus.

© International Software Testing Qualifications Board

Revision History

Version	Date	Remarks
1.0	2026/10/30	Final version for GA approval

© International Software Testing Qualifications Board

Table of Contents

Copyright Notice 2
Revision History 3
Table of Contents 4
Acknowledgements 6
0 Introduction 7
0.1 Purpose of this Syllabus 7
0.2 The Security Test Analyst in Software Testing 7
0.3 Career Path for Testers 8
0.4 Business Outcomes 8
0.5 Learning Objectives and Cognitive Level of Knowledge 8
0.6 The Security Test Analyst Certificate Exam 9
0.7 Accreditation 9
0.8 Handling of Standards 9
0.9 Level of Detail 9
0.10 How this Syllabus is Organized 10
1 Assets and Their Security Risks – 105 minutes (K3) 11
1.1 Asset Security 11
1.1.1 Understanding an asset 11
1.1.2 Value of an asset 13
1.1.3 Common security risks associated with assets to be protected 14
1.2 Security Risk Assessment of Assets 15
1.2.1 Interpreting risk assessment for assets 16
2 Understanding Attacker's Mindset – 105 minutes (K3) 18
2.1 The Attacker Mindset and Their Methods of Operations 18
2.1.1 Motivation of an attacker 18
2.1.2 Using attacker approaches for security testing 19
2.2 Insider and External Attacker 20
2.2.1 Identifying specific risks posed by insider 20
2.2.2 Insider and external attacker in security testing 21
3 Understand Relations Between Risk Management, Control Systems and Regulations – 270 minutes (K4) 23
3.1 Risk Management 23
3.1.1 Risk management methodologies 23

© International Software Testing Qualifications Board

3.1.2 Selecting a risk management methodology 25
3.2 Internal Control Systems 27
3.2.1 The relationship between policies and internal control systems (ICS) 27
3.2.2 Applying security control activities 28
3.3 Defining Test Activities from Regulations 29
3.3.1 Using regulations to derive test activities 29
4 Defining Test Objectives, Security Test Strategy and Milestones – 135 minutes (K3) 32
4.1 Security Test Objectives and Security Test Strategies 32
4.1.1 Test objectives and test strategies for security testing 32
4.1.2 Applying appropriate test strategies for security testing 33
4.2 Security Test Activities 35
4.2.1 Implementing Suitable Security Test Activities within Agile and Waterfall-Based SDLCs . 35
5 Specifying Work Packages for the Security Test Analyst – 150 minutes (K3) 38
5.1 Defining Security Test Work Packages 38
5.1.1 Characteristics of software testing work packages 38
5.1.2 Specifics of security test work packages 39
5.2 Creating a Security Test Plan based on Security Test Work Packages 41
5.2.1 Different aspects involved in security test planning 41
5.2.2 Implementing a test plan for security work packages 42
6 Interpreting Test Results and Defining Actions – 105 minutes (K4) 44
6.1 Interpreting Test Results 44
6.1.1 Key aspects of test report communication 44
6.1.2 Managing security test report follow-ups for vulnerability mitigation 45
6.2 Acting Upon Test Results 46
6.2.1 Analyzing security test reports to prioritize necessary actions 46
7 References 48
Standards 48
ISTQB® Documents 48
Glossary References 48
Books, Articles and Web Pages 48
8 Appendix A – Learning Objectives/Cognitive Level of Knowledge 51
9 Appendix B – Business Outcomes traceability matrix with Learning Objectives 53
10 Appendix C – Security Test Domain Specific Terms 56
11 Appendix D – Trademarks 57
12 Index 58

© International Software Testing Qualifications Board

Acknowledgements

This document was formally released by the General Assembly of the ISTQB® on 2026/10/30

It was produced by a team from the International Software Testing Qualifications Board: Dr. Frank Simon (chair), Beata Karpinska, Maria Kispal, Ralf Pichler, Frans Dijkman, Chris-Julien Becker

The following people participated in the reviewing, commenting and balloting of this syllabus:

David Janota, Earl Burba, Frank Neirynck, Gary Mogyorodi, Ibrahim Anifowoshe, Ioannis Symeonidis, Judy Mckay, Kamila Kozakovičová, Korkor Mensah, Laura Albert, Meile Posthuma, Olubukola Omotayo, Orosz Dénes, Petr Žáček, Ray Gillespie, Taz Daughtrey, Aneta Derkova, Jan Sabak, Jaroslaw, Hryszko, Michael Pilaeten, Nancy Thompson, Graham Bath, Ian Howles and members of the Glossary Working Group, Samuel Ouko and members of the Examination work Group

© International Software Testing Qualifications Board

0 Introduction

0.1 Purpose of this Syllabus

This syllabus forms the basis for the International Software Testing Qualification for the Security Test Analyst (STA). The ISTQB® provides this syllabus as follows:

1. To member boards, to translate into their local language and to accredit training providers. Member boards may adapt the syllabus to their particular language needs and modify the references to adapt to their local publications.
2. To certification bodies, to derive examination questions in their local language adapted to the learning objectives for this syllabus.
3. To training providers, to produce courseware and determine appropriate teaching methods.
4. To certification candidates, to prepare for the certification exam (either as part of a training course or independently).
5. To the international software and systems engineering community, to advance the profession of software and systems testing, and as a basis for books and articles.

0.2 The Security Test Analyst in Software Testing

Security Test Analysts (STA) and Security Test Engineers (STE) play critical roles in ensuring that systems, applications, and organizations are protected against cyber threats. Their main objective is to identify vulnerabilities, weaknesses, and potential attack paths before they can be exploited by malicious attackers.

Instead of using the term “security vulnerability” in this syllabus we only use the term ”vulnerability” for readability improvement !.

The Security Test Analyst qualification is aimed at anyone involved in software testing and has an interest in information security. This Security Test Analyst qualification is also appropriate for anyone who wants a basic understanding of information security from a management perspective.

The Security Test Analyst and Security Test Engineer work closely together when securing information security for identified assets. The analyst uses the business context to define and specify the security test to be performed whereas the engineer is the technical resource executing the security test and reporting their results.

A **Security Test Analyst** focuses on designing, planning, and evaluating security tests based on risk, requirements, and known threat models. They analyze systems from an attacker’s perspective, define test strategies, and interpret the results to assess the overall security posture.

A **Security Test Engineer** is technically focused and is responsible for executing security tests. This includes performing activities such as vulnerability scanning, penetration testing, and controlling security controls. They simulate real-world attacks to uncover technical weaknesses in systems and networks.

Together, Security Test Analysts and Security Test Engineers combine analytical thinking and technical expertise to ensure comprehensive security testing. Their work helps organizations strengthen defenses, improve security awareness, and reduce the risk of cyberattacks.

© International Software Testing Qualifications Board

The Security Test Analyst role is comparable with the following roles:

- Test Analyst: ISTQB® Certified Tester Test Analyst [ISTQB-ALTA]
- Technical Test Analyst: ISTQB® Certified Tester Technical Test Analyst [ISTQB-ALTTA]
- Test Manager: ISTQB® Certified Tester Test Manager Advanced Level [ISTQB-ALTM] and ISTQB® Certified Tester Test Manager Expert Level (Strategic) [ISTQB-ELTM-SM]

0.3 Career Path for Testers

The ISTQB® scheme provides support for testing professionals at all stages of their careers, offering both breadth and depth of knowledge. Individuals who achieve the ISTQB® Security Test Analyst certification may also be interested in acquiring the Security Test Engineer [ISTQB-STE] certification.

0.4 Business Outcomes

This section lists the Business Outcomes expected of a candidate who has achieved the Security Test Analyst certification.

A Security Test Analyst Certified Tester can…

STA-BO1	Explain assets and their related security risks
STA-BO2	Understand Attacker mindset
STA-BO3	Understand relations between risk management, control systems and regulations
STA-BO4	Define test objectives, Security Test Strategy and Activities
STA-BO5	Specify work packages for the Security Test Analyst
STA-BO6	Interpret test results and define actions

0.5 Learning Objectives and Cognitive Level of Knowledge

Learning Objectives support business outcomes and are used to create the Security Test Analyst exams.

In general, all contents of this syllabus are examinable, except for the Introduction and Appendices. The exam questions will confirm knowledge of keywords at K1 level (see below) and of learning objectives at all K-levels.

The specific learning objectives levels are shown at the beginning of each chapter, and are classified as follows:

- K1: Remember
- K2: Understand
- K3: Apply
- K4: Analyze

Further details and examples of learning objectives are given in Appendix A.

© International Software Testing Qualifications Board

For all terms listed as keywords immediately beneath the chapter headings, the correct name and definition from the ISTQB® glossary shall be remembered (K1), even if not explicitly mentioned in the learning objectives.

0.6 The Security Test Analyst Certificate Exam

The Security Test Analyst Certification exam will be based on this syllabus. Answers to exam questions may require the use of material based on more than one section of this syllabus. All sections of the syllabus are examinable, except for the Introduction and Appendices. Standards and books are included as references, but their content is not examinable, beyond what is summarized in the syllabus itself from such standards and books.

Refer to the Exam Structures and Rules document for the Security Test Analyst for further details.

The entry criteria for taking the Security Test Analyst exam is that the ISTQB® Foundation Level certificate is obtained before before taking the Security Test Analyst certification exam.

Other recommendations are:

- that candidates have an interest in both software testing and security testing,
- have at least a background in either software development, such as six months' experience as a system or user acceptance tester or as a software developer,
- have taken a course that has been accredited to ISTQB® standards by an ISTQB-recognized member board.

0.7 Accreditation

An ISTQB® member board may accredit training providers whose course materials follow this syllabus. Training providers should obtain accreditation guidelines from the member board or the body that performs the accreditation. An accredited course is recognized as conforming to this syllabus and can include an ISTQB® exam as part of the course.

The accreditation guidelines for this syllabus follow the general Accreditation Guidelines published by the ISTQB® Processes Management and Compliance Working Group.

0.8 Handling of Standards

Standards associated with quality characteristics and software testing, such as the IEEE and ISO standards, are referenced in the Foundational Level syllabus. The purpose of these references is to provide a framework (as in the references to ISO 25010 regarding quality characteristics) or to provide a source of additional information if desired by the reader. Please note that these standard documents are used for reference only and are not intended for examination.

0.9 Level of Detail

The level of detail in this syllabus enables internationally consistent courses and exams to be provided. To achieve this goal, the syllabus consists of:

- General instructional objectives describing the intention of the Security Test Analyst Level.
- A list of terms that students must be able to recall.
- Learning objectives for each knowledge area, describing the cognitive learning outcome

© International Software Testing Qualifications Board

to be achieved.

- A description of the key concepts, including references to sources such as accepted literature or standards.

The syllabus content is not a description of the entire knowledge area of security testing; it reflects the level of detail to be covered in Security Test Analyst Level training courses.

The syllabus uses the terminology (i.e., the name and meaning) of the terms used in the testing and quality assurance of software and security according to the ISTQB® Glossary.

0.10 How this Syllabus is Organized

There are six chapters with examinable content. The top-level heading for each chapter specifies the time for the chapter. Timing is not provided below chapter level. For accredited training courses, the syllabus requires a minimum of **14,5** hours of instruction, distributed across the chapters as follows:

- Chapter 1: 105 minutes Assets and Their Security Risks
- Chapter 2: 105 minutes Understand Attackers' Mindset
- Chapter 3: 270 minutes Understand Risk Management, Control Systems and Regulations
- Chapter 4: 135 minutes Define Test Objectives, Security Test strategy and Milestones
- Chapter 5: 150 minutes Specify Work Packages for the Security Test Engineer
- Chapter 6: 105 minutes Interpret Test Results and Define Actions

© International Software Testing Qualifications Board

1 Assets and Their Security Risks – 105 minutes (K3)

Keywords

risk appetite

Domain Specific Keywords

asset, asset inventory, incident

Learning Objectives for Chapter 1:

1.1 Asset Security

STA-1.1.1 (K2) Explain what an asset is

STA-1.1.2 (K2) Explain the value of an asset

STA-1.1.3 (K3) Apply risk identification techniques to determine potential security risks to a protected asset

1.2 Security Risk Assessment of Assets

STA-1.2.1 (K2) Interpret risk assessment for assets

1.1 Asset Security

Security is not a standalone characteristic. It only has meaning in the context of the assets it is intended to protect. Therefore, any discussion of security must begin with a clear understanding of the assets, their value, and the risks they face.

1.1.1 Understanding an asset

The term “asset” is used to describe any data, device, system, or component that supports information-related activities and has value to the organization requiring protection. In the context of information security management, assets include information itself as well as related processes, systems, networks, people, and facilities that are important for achieving organizational objectives [ISO 27000]. Assets are items of value to achieve organizational mission/business objectives [NIST-Glossary].

The identification of assets is a time-consuming task which defines a foundation for all security activities. Identification can be based on experience gained from previous projects, use of tools, performing brainstorming sessions, and conducting discussions with experts.

Assets may be classified as either tangible or intangible assets.

- Tangible assets have a physical form which can be touched or handled. This includes:
  - Hardware assets, such as computer systems, data storage, and data communication devices.
  - Physical security assets, such as locks and security cameras.

© International Software Testing Qualifications Board

- Intangible assets are non-physical and have value due to their intellectual, informational, or legal characteristics. This includes:
    - Software assets, such as operating system, applications, and databases.
        - Information assets, such as business plans, documentation, inventions, pictures, and personal records.
        - Personal assets, such as personal identifiable information (PII) and biometric data.
        - Intellectual property (IP) assets, such as patents, copyrights, trademarks, and trade secrets.
        - Service and cloud assets, such as hosting platforms, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), and containerized environments.
        - Reputational assets, such as brand, image, customer loyalty, and trust.

Assets may be further categorized according to the following aspects:

- Data Sensitivity: Data can be classified according to its sensitivity, ranging from public to confidential to highly classified. Sensitive data requires stricter security measures to protect it from unauthorized access, disclosure, alteration, or destruction.
- Access Level: Assets can be categorized according to who has access to them. Public assets are accessible to anyone, while private assets require authorized access. Sensitive assets may require restricted access to authorized personnel only.
- Criticality: Assets can be categorized according to the degree to which the organization's business operations are impacted. For example, critical assets require robust security measures to prevent disruptions or damage.
- Assets are typically categorized using intrinsic properties such as data sensitivity, criticality to business continuity, legal/compliance obligations, and asset type (e.g., on-premise, cloud services, third-party components).
- Security Controls: Assets can be categorized according to the security controls implemented to protect them, such as firewalls, intrusion detection systems, access control measures, and data encryption.
- Regulations: Assets can be categorized according to the applicability of industry-specific regulations, compliance requirements, and regulations driven by authorities. This verifies that security measures align with the applicable regulatory frameworks.

For each asset, security measures are implemented to uphold the CIA triad – Confidentiality, Integrity, and Availability – thereby protecting them from unauthorized access, use, alteration, and destruction [ISTQB-STE]. Confidentiality protects information from unauthorized disclosure. Integrity ensures information remains accurate and unaltered. Availability ensures information and systems are accessible when needed. Note that attacks may primarily impact only one attribute. Equally important is the AAA framework (Authentication, Authorization, and Accounting), which complements CIA by verifying user identity, defining permitted actions, and maintaining auditable records for both physical and intangible assets.

All assets within the scope of risk analysis must be identified and documented in an asset inventory. This inventory should include each asset's category, value, owner(s), and other relevant attributes for accountability. The inventory must be reviewed and updated regularly.

To address supply chain risks, maintaining a Software Bill of Materials (SBOM) is recommended for software assets. An SBOM provides a formal inventory of components and dependencies to improve vulnerability tracking.

The asset inventory list must be updated under the following circumstances:

- new assets have been identified,
- an asset is decommissioned or disposed of,

© International Software Testing Qualifications Board

- the attributes of an asset (e.g., business value, ownership, or purpose) have changed.

If the assets are not managed properly during their lifecycle, the organization may become unaware of certain risks. For example, outdated software may expose vulnerabilities, or improperly disposed of hardware could lead to data leaks. These oversights can result in security breaches, compliance failures, and financial loss.

The asset inventory list itself is also an asset and must be kept adequately protected (e.g., with access control, encryption, physical security, and regular audit). The ongoing activity of identifying, classifying, evaluating, and maintaining records of assets plays a fundamental role within multiple organizational frameworks, including Business Continuity Management Systems (BCMS) and Information Security Management Systems (ISMS). They provide essential information for risk management, resource allocation, incident response, compliance, and overall organizational resilience.

An asset is considered to be relevant to security if there is at least one damage scenario that impacts one or more core security goals. Damage scenarios are handled as part of risk management.

1.1.2 Value of an asset

All assets have value. Some are quantified by the consequences of their loss, while others may not be as easy to measure. The distinction often lies in the nature of the assets and their impact on an individual or organization.

An asset's value can be expressed using quantitative as well as qualitative scales. A quantitative value usually refers to the asset's monetary value, whereas a qualitative value classifies an asset's value on a graded scale such as high, medium, or low. The ability to quantify the loss of an asset can vary, and a holistic approach to risk management must be adopted.

Some of the factors to consider when estimating the value of an asset include:

- The time and effort needed to recreate or recover the asset.
- Potential damage caused by loss of the asset.
- The potential future revenue to be generated by the asset.
- Penalties (e.g., for data leaks, violation of data privacy, non-disclosure of assets).
- Legal liabilities.
- Business continuity impact.
- Compliance costs.
- Reputational damage and loss of customer trust.
- Loss of competitive advantage (e.g., intellectual property, trade secrets).

Tangible assets can often be quantified by considering their replacement cost or depreciated value. However, in security contexts, the valuation of tangible IT assets such as servers, networking equipment and specialized hardware also requires the consideration their role in critical operations, the sensitivity of the data they process or store, potential regulatory impacts if compromised, and the operational disruption costs beyond their market value.

Intangible assets are more difficult to measure in monetary terms. However, there are methods that can be used to estimate the value of intangible assets, such as:

- Discounted Cash Flow (DCF) analysis: This calculates the current value of expected future cash flow using a discount rate.
- Market valuation: This estimates value based on the prices paid for similar assets on the market.
- Cost-to-replace analysis: This estimates the cost of recreating the intangible asset from scratch.

© International Software Testing Qualifications Board

- Quantitative risk assessment methods, particularly Single Loss Expectancy (SLE), Annualized Rate of Occurrence (ARO), and Annualized Loss Expectancy (ALE), which are widely used in security risk management to calculate expected monetary loss.

1.1.3 Common security risks associated with assets to be protected

Assets become vulnerable to risk when threats can exploit weaknesses that affect their confidentiality, integrity, or availability. For successful risk management, assets must first be identified.

A key tool for understanding the consequences of these risks is the Business Impact Analysis (BIA). In a security context, BIA specifically evaluates the operational, financial, and continuity impacts of security failures on critical assets, helping organizations prioritize mitigation and preparedness.

A related but distinct process is the Privacy Impact Assessment (PIA), also known as Data Protection Impact Assessment (DPIA) under GDPR Article 35. While BIA primarily addresses operational and financial impacts, a DPIA specifically assesses the risks to individuals' privacy rights and freedoms resulting from personal data processing activities. DPIAs are legally required for high-risk processing operations and are essential for compliance in GDPR-regulated environments.

In many industries, compliance with data privacy regulations such as General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA) requires organizations to conduct BIAs. These assessments demonstrate the organization's commitment to data protection and its understanding of the potential risks associated with data breaches.

BIAs play a critical role in helping organizations to:

- Identify and prioritize critical assets and processes.
- Assess the potential impacts of security failures.
- Prioritize risk mitigation strategies.
- Develop recovery plans.
- Demonstrate compliance with data privacy regulations.

Identifying assets to be protected cannot usually be done for an individual asset without considering dependencies on other assets. Without such consideration, there is a risk of missing an unforeseen chain of events.

It is important to include the relevant stakeholders in evaluating assets as they are the ones with insights into the system and are able to determine asset types, asset impacts, and the required level of protection necessary for each asset.

Based on the previous classification of tangible and intangible assets, the following lists provide examples of their associated security risks:

For tangible assets:

- Physical theft
- Damage or destruction
- Loss or misplacement
- Cyberattacks

For intangible assets:

- Unauthorized access
- Data breaches
- Malware
- Social engineering
- Distributed Denial of Service (DDoS) attacks
- Zero-day exploits

© International Software Testing Qualifications Board

- Non-compliance risks
- Supply chain attacks

It is important to distinguish between threats and security events. Common threats to intangible assets include malicious software (malware) and zero-day exploits. These threat vectors can trigger security events such as unauthorized access or DDoS attacks, ultimately resulting in impacts like data breaches or regulatory non-compliance [ISO 27005].

To protect assets from these typical risks, organizations can implement a variety of security measures.
For tangible assets, typical security measures are:

- Physical Access Controls: Use of badge readers, biometric scanners, and mantraps.
- Environmental Controls: Installation of fire suppression systems, temperature and humidity sensors, and uninterruptible power supplies (UPS) to protect physical assets from damage.
- Asset Tagging and Inventory Management: Use RFID tags or barcode systems to track physical assets and reduce loss or misplacement.
- Surveillance and Monitoring: Deploy CCTV cameras and motion detectors around physical asset locations to prevent theft and vandalism.
- Secure Disposal Procedures: Implement strict protocols for the disposal or recycling of physical equipment to prevent unauthorized data recovery.
- Industrial Control Systems Protection: Application of network segmentation, hardware firewalls, etc.

For intangible assets, typical security measures are:

- Role-Based Access Control (RBAC): Enforcing strict permissions so that only authorized personnel can access sensitive data or critical systems.
- Cryptographic methods: Encryption of sensitive data at rest and in transit using strong cryptographic algorithms to protect confidentiality. Applying hashing and digital signatures to support data integrity and authenticity.
- Endpoint Protection and Anti-Malware: Deploy advanced endpoint security solutions that include behavior-based detection to identify and block malware and Zero-Day exploits.
- Security Awareness Training: Conduct regular training sessions to educate employees about social engineering tactics and phishing threats.
- DDoS Mitigation Services: Use cloud-based or on-premise DDoS protection tools that detect and absorb traffic spikes to maintain service availability.
- Vulnerability Management and Patch Management: Implement continuous scanning and timely patching of software to reduce exposure to Zero-Day and known exploits.
- Incident Response Planning: Develop and regularly test incident response and disaster recovery plans to support rapid detection, containment, and recovery from security events.
- Compliance Management Tools: Utilize software to monitor regulatory requirements, automate audits, and maintain evidence of compliance to reduce legal risks.
- Data Loss Prevention (DLP) Solutions: Apply to DLP policies to monitor, detect, and block unauthorized transmission of sensitive information via email, USB devices, or cloud services.

Incident response planning outlines the steps that an organization will take to respond to a cybersecurity incident.

1.2 Security Risk Assessment of Assets

Risk assessment plays a crucial role in proactively identifying and managing security and operational risks during system development. It is an ongoing activity throughout the System Development Life Cycle (SDLC), from requirements gathering to maintenance.

© International Software Testing Qualifications Board

Risk assessment is the overarching process that includes risk identification, risk analysis (evaluating likelihood and impact using the CIA triad), and risk evaluation. It is distinct from Business Impact Analysis (BIA), which focuses on the operational and financial consequences of disruptions to critical assets.

1.2.1 Interpreting risk assessment for assets

Risk assessment is the process of identifying, analyzing, and evaluating security risks, including likelihood and potential impact on confidentiality, integrity, and availability, against the organization's business processes.

Business impact analysis (BIA) looks at how disruptions can affect important business processes and the organization. Risk assessment is about finding and understanding specific threats and weaknesses that could harm the organization.

The two processes are linked in different ways:

- Risk assessment provides input to BIA. The results of a risk assessment can be used to identify the risks that are most likely to disrupt critical business processes. This information can then be used to prioritize the processes that need to be included in the BIA.
- BIA provides input for risk assessment. The findings of a BIA can be used to identify new risks that may not have been considered in the initial risk assessment. This information can then be used to update the risk assessment and maintain it in an accurate and up-to-date condition.

Risk assessment and BIA contribute to business continuity planning. The results of both processes can be used to develop effective business continuity plans that will protect critical assets so that the organization can resume operations quickly in the event of a disruption.

Key steps involved in a security risk assessment for assets are:

Step	Activity
Asset Identification	Identify and categorize all the assets within the organization based on the scope and objectives.
Asset Valuation	Assign a value to each identified asset.
Threat Identification	Identify and assess potential threats that could exploit vulnerabilities in the organization's assets.
Vulnerability Assessment	Identify and assess vulnerabilities associated with each asset.
Risk Analysis	Evaluate the risk likelihood and potential risk impact of identified threats exploiting vulnerabilities.
Risk Prioritization	Prioritization of the identified risk based on the criticality of the assets they may affect.
Risk Treatment	Select and implement appropriate risk treatment options (e.g., mitigation through security controls, risk transfer, risk avoidance, or risk acceptance) to appropriately manage the risk to the assets.
Continuous Monitoring	Regularly review and update the asset inventory, valuation, and risk assessment.

© International Software Testing Qualifications Board

Quantitative risk assessment uses mathematical models to estimate the risk likelihood and risk impact of potential cyber threats. It is often employed for more complex assessments or when detailed data is available, and a higher degree of precision is desired. While quantitative methods are generally perceived as being more objective than qualitative approaches, they can still be influenced by biases introduced through inaccurate data or flawed assumptions. It does not automatically guarantee objectivity.

Qualitative risk assessment is a subjective approach that relies on expert judgment to assess the risk likelihood and risk impact of potential cyber threats. It is often used for high-level assessments or when there is limited data available.

A risk matrix is commonly used in qualitative risk assessment. It is a visual representation that allows organizations to prioritize their efforts and resources. The matrix typically consists of a grid with risk likelihood on one axis and risk impact on the other. The risk likelihood is estimated for all identified assets and threat combinations.

Risk likelihood is determined by:

- The probability or frequency of a threat event occurring, and
- The probability of successful exploitation, considering factors such as attach complexity, exposure, and required privileges.

While traditional qualitative risk matrices using impact and probability provide a foundational framework, they can be subjective and may not fully account for complex scenarios. Advanced quantitative methodologies, such as Factor Analysis of Information Risk (FAIR ™), build upon these concepts by using event frequencies, probability ranges, and financial modeling to provide a more rigorous, business-aligned risk estimation. [FAIR ™]

Risks with a high likelihood and high impact typically exceed the organizational risk appetite and require immediate prioritization. The exact timeframe for implementing initial mitigation actions and developing a formal risk treatment plan is governed by the organization's specific risk management policies and incident response SLAs.

This prioritization of risks by risk likelihood and risk impact is a key step in the risk management process, which involves identifying, assessing, prioritizing, and treating risks to align with business objectives and legal or regulatory requirements.

Within the risk management process, organizations establish a risk ‘appetite’, which relates to an organization’s willingness to accept and tolerate risks associated with its information technology systems and data. It represents the risk level that an organization is prepared to accept to achieve its business objectives. Determining and defining risk appetite involves assessing the potential risk impact of security failures and the risk likelihood of their occurrence.

The risk appetite is then communicated through policies, training, and integration into business processes by the Board, senior management, and risk managers so that everyone understands and applies to the acceptable risk level in daily decisions. Risk appetite statements are clear and concise statements that articulate the organization's stance on risk. They provide guidance in the decision-making process.

Risk tolerance is the acceptable level of variation an organization is willing to tolerate in achieving its objectives. It helps in defining specific thresholds for risk that the organization is not willing to exceed.

Risk thresholds set specific limits or thresholds for various types of risks. These thresholds help identify when the risk level is becoming unacceptable, triggering the need for risk mitigation measures.

© International Software Testing Qualifications Board

2 Understanding Attacker's Mindset – 105 minutes (K3)

Keyword

phishing, reconnaissance

Domain Specific Keywords

active reconnaissance, attacker, attack surface, baiting, deepfake, dumpster diving, insider threat, MITRE ATT&CK, Open-Source Intelligence, passive reconnaissance, QR phishing, Quid Pro Quo**,** scareware, smishing, social engineering, spoofing, watering hole attacks

Learning Objectives for Chapter 2:

2.1 Attacker mindset and their methods of operations

STA-2.1.1 (K2) Give examples of how knowledge of an attacker's motivations can enhance the work of a Security Test Analyst

STA-2.1.2 (K3) Apply methods an attacker might use to discover key information that can be leveraged in security testing

2.2 Insider and external attacker

STA-2.2.1 (K2) Give examples of specific risks posed by an insider.

STA-2.2.2 (K2) Distinguish between external attackers and insiders in the context of security testing

2.1 The Attacker Mindset and Their Methods of Operations

Understanding what motivates attackers is essential for anticipating their actions and strengthening security defenses. Whether driven by financial gain, ideology, or opportunity, these motivations influence how attacks are planned and executed. By recognizing these drivers, Security Test Analyst can better identify high-risk assets and tailor test strategies to address the most relevant threats.

2.1.1 Motivation of an attacker

Organizations face threats from a diverse range of attackers, each with different objectives, skill levels, and available resources. For Security Test Analyst understanding attackers' motivations, objectives, and capabilities is essential for realistic threat modeling and effective security testing. This understanding helps in developing security test conditions and test cases that address relevant threats, identify exploitable vulnerabilities, and support stakeholders in mitigation of security risks.

The most common motives for attacking are described below [Morpus22] [Cyber]:

- **Financial Gain**
  This is one of the most common motivations for cyberattacks. Attackers employ various methods to profit from their activities, such as demanding ransom in exchange for stolen data, selling sensitive information on the dark web, or directly stealing money through compromised credit cards or bank accounts.
- **Espionage and Sabotage**

© International Software Testing Qualifications Board

Sabotage and espionage are major drivers for nation-state actors or corporate spies. These actors conduct targeted cyber espionage to obtain sensitive information, including intellectual property, strategic research, and national security secrets. Their objectives often extend beyond information theft, aiming to undermine an organization's reputation, disrupt critical operations, or compromise critical infrastructure.

- **Political Statements**
  These attackers, often referred to as "hacktivists". Hacktivists are driven by ideological or political beliefs. For example, a group might hack and take down a website to protest against political decisions or social injustice.
- **Revenge**
  Disgruntled employees or former partners sometimes turn to cybercrime to seek revenge.
- **Power and Control**
  Many attackers are motivated by power and recognition rather than money. They enjoy demonstrating their technical abilities, overcoming security barriers, and gaining control over systems or data. Successfully manipulating systems or accessing restricted information can give them a sense of power and achievement.

2.1.2 Using attacker approaches for security testing

An attacker typically begins by performing reconnaissance to gather information about the target system, infrastructure, organization, or users. Reconnaissance may involve both technical exploration and social engineering activities. The information collected can be used to identify potential attack paths, vulnerabilities, and valuable assets.

Reconnaissance techniques are described in the MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework which is a globally accessible knowledge base of adversary tactics and techniques based on real-world cyber intrusions. It provides a structured way to understand how attackers operate, from their initial reconnaissance activities to the execution of malicious actions within a target environment [MITRE].

Reconnaissance includes both passive and active reconnaissance techniques. Passive reconnaissance gathers information without interacting with the target, making it stealthy and difficult to detect. It often relies on Open-Source Intelligence, OSINT [OSINT] sources such as websites, social media, public records, WHOIS databases, search engines, and publicly accessible cloud storage. Techniques like dumpster diving may expose discarded documents containing sensitive information that attackers could use when planning an intrusion [Wright, 2021].

Active reconnaissance involves direct interaction with the target, increasing the likelihood of detection. Typical methods include pinging, port scanning, and probing network defenses to map the attack surface.

By applying attacker thinking, the security test analyst interprets technical exploration results to identify threats, attack surface, and security risks. This enables risk-based security testing, where test conditions are prioritized and focused on the most critical assets and realistic attack scenarios.

The growing use of artificial intelligence (AI) is transforming both reconnaissance and security testing activities. AI-powered tools can rapidly collect, analyze, and correlate information from multiple sources, helping security test analyst identify potential vulnerabilities, exposed assets, and attack paths more efficiently [AI26].

Social engineering is the use of psychological manipulation to influence individuals into disclosing sensitive information or performing actions that compromise security. It exploits human factors such as trust, helpfulness, fear, and respect for authority. Attackers use social engineering to obtain

© International Software Testing Qualifications Board

confidential information, bypass security controls, gain unauthorized access, or induce users to install malware or perform other actions that support an attack [BSI24].

The most common types of social engineering attacks include [EveSec]:

- Phishing – deceptive emails crafted to steal credentials or spread malware.
- Baiting – using physical or digital bait to lure victims.
- Quid Pro Quo – offering a service or reward in exchange for confidential data.
- Scareware – creating fear through fake warnings or threats to pressure users into taking impulsive actions.
- Watering Hole Attacks – compromising trusted websites to target visitors.
- Business Email Compromise – impersonating executives to authorize fraudulent transactions.
- DNS Spoofing/Cache Poisoning – redirecting traffic to malicious websites by corrupting DNS data.
- Tailgating – deceiving an employee to gain physical entry by following an authorized person into a restricted area without having the required access rights.
- Other methods including deepfakes, SMS phishing (smishing), QR-codes phishing (QR phishing), and social media-based manipulations.

Security Test Analyst can use social engineering to assess the effectiveness of security controls and identify human-related vulnerabilities. By simulating attack scenarios, analysts can evaluate users' ability to recognize and respond to social engineering attacks and provide information for improving security awareness and risk mitigation actions.

2.2 Insider and External Attacker

Understanding both insider and external attackers is a crucial part of cybersecurity. An external attacker is an unauthorized entity outside the security domain that poses a potential threat to an information system. An insider is often referred to as an insider threat, is an individual who misuses authorized access to systems, applications, or information assets [NIST-Glossary].

2.2.1 Identifying specific risks posed by insider

One of the most frequently underestimated threats to an organization comes from insiders. While external threat actors, such as cybercriminal, are responsible for most security breaches [Verizon24], insider threats remain a significant risk which may lead to extensive harm to both an organization's operation and its reputation.

An insider threat arises when an individual with authorized access such as an employee, contractor, or partner, misuses that access, intentionally or unintentionally, to compromise the organization's systems, networks, or information assets [Poremba24]. Insider actions may involve espionage, sabotage, unauthorized disclosure of information, or the loss or degradation of organizational resources and capabilities. These risks arise because insiders already possess trusted access, knowledge of systems, and awareness of security controls, making detection more challenging, and the potential impact significantly higher.

© International Software Testing Qualifications Board

Insider threats can be categorized as [CISA]:

- **Unintentional threats** – Security failures caused by human error, negligence, or failure to follow security policies.
- **Malicious threats** – Individuals who intentionally misuse authorized access to steal data, commit fraud, sabotage systems, or disclose confidential information.
- **Collusive threats** – Insiders who cooperate with external threat actors to facilitate unauthorized activities.
- **Third-party threats** – Contractors, vendors, or partners with authorized access who may introduce security risks, intentionally or unintentionally.

From a security perspective, insiders pose a high-risk level due to both technical and social factors.

Typical technical factors are:

- Insiders have authorized accounts and legitimate access to systems, applications, and data.
- They often use company-managed devices configured with corporate certificates, credentials, and authentication mechanisms.
- They operate within the organizational network and trusted environment, potentially bypassing perimeter security controls.

Typical social factors are:

- Insiders are trusted individuals, which can cause suspicious behavior to be overlooked or misinterpreted.
- Requests from internal users are often perceived as legitimate, increasing the likelihood that information will be shared without proper verification.
- Established relationships and familiarity with the organization make social engineering attacks easier to execute.

2.2.2 Insider and external attacker in security testing

It is important for security test analyst to distinguish between insider and external threats. The main dimensions to consider are:

- Scope and focus
  - Insider threat testing focuses on risks from users with authorized access. It examines both malicious and unintentional actions that could compromise organizational assets, including misuse of privileges, policy violations, and negligent behavior.
  - External threat testing evaluates vulnerabilities in external facing systems (e.g., web applications, Application Programmable Interface (API)). It simulates attackers operating outside the organization with no authorized access.
- Attack surface
  - Insider threat testing includes systems, data flows, and processes accessible to authorized users. Testing focuses on access controls, internal data handling, privilege boundaries, and interactions between internal systems.
  - External threat testing includes all entry points exposed to the internet or external networks, such as firewalls, web servers, public facing applications, and external APIs. Testing evaluates perimeter security and externally exploitable weaknesses.

© International Software Testing Qualifications Board

- Authorization and context
  - Insider threat testing assumes the attacker already has some level of legitimate access. Testing simulates actions possible within assigned roles, trust levels, and privileges.
  - External threat testing assumes no prior authorization. Testing reflects an attacker with no internal knowledge attempting to breach defenses from outside.
- Types of attacks
  - Insider threat testing includes misuse of privileges, data theft, privilege escalation, sabotage, unauthorized access to sensitive information, and internal social engineering. Testing focuses on weak access controls, excessive privileges, and monitoring gaps.
  - External threat testing includes attacks such as SQL injection, cross-site scripting, brute force authentication, exploitation of exposed services, and other vulnerabilities accessible from outside the organization.

Security Test Analyst should assess both insider and external threats to determine the appropriate testing focus. Test results help identify security improvements, such as strengthening access controls, monitoring, and incident response for insider threats, or enhancing patch management, perimeter defenses, and secure coding practices for external threats.

© International Software Testing Qualifications Board

3 Understand Relations Between Risk Management, Control Systems and Regulations – 270 minutes (K4)

Keywords

risk appetite, risk management

Domain Specific Keywords

asset, business impact analysis, internal control system

Learning Objectives for Chapter 3:

3.1 Risk Management

STA-3.1.1 (K2) Explain risk management methodology

STA-3.1.2 (K4) Select a risk management methodology for identified security risks

3.2 Internal Control Systems

STA-3.2.1 (K3) Apply the relation between policies and internal control systems

STA-3.2.2 (K3) Apply security control activities

3.3 Test activities from Regulations

STA-3.3.1 (K3) Use security regulations to derive test activities

3.1 Risk Management

Once the relevant assets have been identified, the next step is to understand the risks that may affect them. Risk management enables organizations to evaluate threats, vulnerabilities, and potential impacts, creating a rational basis for security decisions.

3.1.1 Risk management methodologies

Active risk management, in which risk assessment is only one step (see section 1.2), supports an actively managed process of handling risks exists: This means that risks are not just ignored or overlooked but continuously monitored and actively decided [ISTQB-Glossary]. Handling risks is not limited to avoiding any risks, but also includes identifying them, evaluating them, and systematically treating them.

ISO 31000 [ISO 31000] provides a generic and well-known risk management process which consists of the following 5 major steps/activities:

Step	Activity
1. Establishing the context	The overall context in which the core business of the organization is defined is identified. This includes:

© International Software Testing Qualifications Board

Step	Activity
	• The objectives of an organization. • An organization's core activities. • People in the organization, both internally and externally (such as freelancers and consultants). • Laws, regulations, rules, or standards that apply to the organization. Possible risks that are outside this context (e.g., political troubles) might have only a very limited effect on the business and don't need any dedicated treatment.
2. Risk assessment	This consists of • **Risk identification** (specified risk at specified location), • **Risk analysis** (establish details of the risk), and • **Risk evaluation** (what does the risk impact for the specific business). The goal of risk assessment is to establish what the effective risk is within a specific context. Usually, standard scales for assessing risk likelihood and risk impact are used, which support the location of every risk in its corresponding risk matrix (see section 1.2.1).
3. Risk treatment	This involves the final treatment of assessed risks. ISO 31000 suggests four different ways to treat a risk. Two important terms for govern different types of risk treatments are risk threshold and risk tolerance, because they define how much risk you accept without doing anything: • **Accept**: If potential opportunities are considered to outweigh the risk, or no other risk treatment is appropriate, the risk can be accepted. The risk itself stays unchanged, although this differs from ignoring the risk, as the acceptance is explicit and based on a systematic process. • **Mitigate**: Reducing one or both of the factors of risk likelihood or impact mitigates the overall risk. Risk mitigation actions must be performed until other risk treatment is possible. In many cases successful risk mitigation ends with accepting the remaining risk. This indicates that the risk is considered acceptable and within an organization's risk appetite. • **Avoid**: Deciding not to commence or to continue with the activity that gives rise to the risk. This generally means that the asset with the associated risk is deleted or moved outside the context so that no further risk likelihood or impact is considered. • **Transfer/Delegate**: Sometimes the impact of a risk can be transferred/delegated, such as by purchasing insurance which covers the negative impact of the risk should it occur.
4. Communication and consultation	While steps one to three are ongoing, continuous communication with the stakeholders and consultation with relevant experts must be performed. For example, many regulated industries must communicate assessed risks above a predefined threshold to the regulatory authority. Sometimes risks must be communicated by the Security Test Analyst to be included into an organization-wide risk management or must be shared with stakeholders.

© International Software Testing Qualifications Board

Step	Activity
5. Monitoring & Reviewing, Continuous process improvement	As risk management is crucial for companies, the process must be continuously monitored, reviewed and – if possible - improved.

An alternative to ISO 31000 is OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation): It is a risk-based information security assessment methodology developed by the Software Engineering Institute (SEI) at Carnegie Mellon University [Alberts03]. OCTAVE focuses on identifying and evaluating risks to an organization's critical assets by considering business objectives, operational practices, threats, and vulnerabilities. The framework emphasizes self-directed assessments conducted by personnel who possess deep knowledge of the organization's processes and systems. As a result, OCTAVE helps organizations understand how information-related risks affect business operations and supports the development of appropriate risk mitigation strategies.

A specific characteristic of OCTAVE is its asset-centric perspective. Rather than beginning with technical vulnerabilities – which is typical for ISO 31.000 – the methodology starts by identifying critical information assets and then analyzes the threats and organizational weaknesses that could compromise them. This business-driven approach has made OCTAVE particularly valuable for organizations seeking to align information security activities with broader enterprise risk management objectives.

While both ISO 31000 and OCTAVE provide structured approaches for managing risk, they differ significantly in scope, purpose, and application.

ISO 31000 is a generic enterprise risk management framework that can be applied to any type of risk, including strategic, operational, financial, compliance, and information security risks. It introduces a governance framework, and a systematic process for identifying, assessing, treating, monitoring, and communicating risks throughout an organization. As such, ISO 31000 serves as an overarching management framework that can be integrated into organizational decision-making at all levels.

OCTAVE, by contrast, was specifically designed for information security risk assessments. It provides a detailed methodology for identifying critical information assets, analyzing threats and vulnerabilities, and evaluating the resulting risks. Whereas ISO 31000 describes what organizations should do to manage risk, OCTAVE provides more detailed guidance on how to perform a security-focused risk assessment.

From a practical perspective, the two approaches are not mutually exclusive. Many organizations use ISO 31000 as their enterprise-wide risk management framework and apply OCTAVE as a specialized method for assessing cybersecurity and information security risks within the broader ISO 31000 process. In this sense, OCTAVE can be viewed as a complementary technique that supports the risk identification, analysis, and evaluation activities defined by ISO 31000.

Two further typical risk management frameworks are FAIR (see 1.2.1) and NIST-Risk management framework (NIST-RMF) [NIST 800-37].

3.1.2 Selecting a risk management methodology

It is important to differentiate between a security incident and a security risk. A security incident covers one or more aspects of the CIA triad, consisting of confidentiality, integrity, and availability. Events such as when data has been stolen, or if a system is not available are not represented by risk management, but by incident management. However incident management and risk management are not completely working in different silos. In best case an incident, i.e. something has really happened,

© International Software Testing Qualifications Board

was already foreseen by risk management. That means, that all information, that were gathered for a holistic and complete risk assessment can and should be used for incident management if the risk happens (i.e. probability is 100%).

A security risk leads to a negative impact (in terms of the CIA triad) if something occurs in the future with a certain risk likelihood.

Security testing is a typical activity used to identify risks and to assess them in more detail. Usually, security testing only makes sense for IT systems that are inside the pre-defined organizational context. Identified risks outside the context are still risks, but their management is outside the organization's scope.

The Security Test Analyst applies security testing for effective risk management. When considering the five tasks described in ISO 31000 (see section 3.1.1 above) this means:

Step	Tasks of the Security Test Analyst
1. Establishing the context	The Security Test Analyst analyzes all IT systems that are relevant for the business inside its context. Usually, the IT systems of an organization have different degrees of relevancy: Some IT systems might be necessary, and any security failures can have severe consequences. Other IT systems might only decrease the efficiency or satisfaction of working, but security failures have only very limited negative risk impact on the business. An important source of information regarding risk assessment is the business-impact-analysis (BIA): A BIA determines the criticality of business activities and associated resource requirements to support operational resilience. Usually, the BIA quantifies the impact of disruptions, like when affected by security failures, on service delivery [Gart23].
2. Risk assessment	Based on the context, the Security Test Analyst defines the desired test objects, test strategies, and coverage for the risk assessment. It's the task of the security test engineer to apply security test techniques to identify risks (e.g., identified vulnerabilities), to provide as many details as possible to the context of the risk analysis, and to support the risk evaluation. If the concept of risk levels are applied, which classify risks in different intervals (e.g. low, medium, high), the security test analyst supports this activity as well. The following steps are part of the risk assessment and are essential for effective risk management: • **Risk identification**: In the first step, the security test engineer identifies risks. In general, risk management can only manage risks that are identified. The Security Test Analyst defines the security test strategy to get an overall risk picture that is as complete as possible. • **Risk analysis**: Not all identified risks are exposing a risk to the business: e.g., not all identified vulnerabilities can be exploited in a certain operational context, and not all weaknesses affect all business processes. It's the responsibility of the Security Test Analyst to understand a security risk and to collect all relevant data as part of this step. The security test engineer supports this with as much information as possible. • **Risk evaluation**: Even if a security test engineer's task is to identify and analyze security risks, they are an important source of information that might further influence the risk treatment. When the risk evaluation is done, the security test engineer's results are enriched by business aspects, like what is the possible impact on the business,

© International Software Testing Qualifications Board

Step	Tasks of the Security Test Analyst
	and what is the worst-case scenario if a security risk occurs. In this step, the business supports the Security Test Analyst with the evaluation. They then explain technical details and understand the potential impact of the business.
3. Risk treatment	The risk treatment is usually done by the executive level of an organization as this is the highest level in an organization who can effectively accept a risk. However, the Security Test Analyst has to hand over all relevant information from the risk assessment to them to allow for a holistic and systematic risk treatment.
4. Communication and consultation	Communication of test results is a very sensitive action (see [ISTQB-STE]). Usually, the level of details differs in terms of criticality of content and of skill level on the audience side. In business organizations risk catalogues, risk assessments and risk treatments are handled on a strictly confidential level. In the consultation activity, the Security Test Analyst can leverage external security resources/experts for very specific items. E.g., if a very specific firewall is identified as essential while setting the organizational context, it might be necessary to identify a security test engineer for this specific device. Other consultations might include getting advice from external security institutions (e.g., MITRE Corporation) or governmental bodies.
5. Monitoring and Reviewing, continuous process improvement	As all systems usually change over time, it is important to monitor these changes and to review the applied activities. If a new IT provider comes in, this might change the overall organizational context and suggest completely new test objects, test objectives, test strategies, and coverage. Sometimes the IT remains unmodified, but the business processes change: One system, which looked irrelevant in a former analysis step, can become very critical when a business-critical process is now run on that system. It's the task of the Security Test Analyst to analyze all changes and to adjust the overall security test strategy.

3.2 Internal Control Systems

Internal Control Systems (ICS) translate management objectives and risk responses into concrete operational and technical controls. In the security domain, these controls are designed to protect information assets, reduce risk, and ensure compliance with organizational requirements.

3.2.1 The relationship between policies and internal control systems (ICS)

Any test activity needs a test basis. Testing itself is never a first step in a development project. To improve the process of creating an asset, policies can be helpful. A security policy can be defined as follows:

> *"a set of ideas or a plan of what to do in particular situations that has been agreed to officially by a group of people, a business organization, a government, or a political party" [Camb23]*

If a security policy is in place for creating, using, modifying or deleting any asset in an organization, it is essential to perform a compliance check on the actions of any assets based on that policy. Any security policy without such checks cannot effectively control whether asset changes reflect the

© International Software Testing Qualifications Board

policy, and no feedback can be given. This constraint applies to security policies as well as test policies as defined in [ISTQB-Glossary].

Security policies exist on different levels of granularity: They may be defined for secure data transfers between two machines, for secure coding, for logging, and for overall data center management.

Together with the previously mentioned constraint that all policies should be supported by test conditions, it is evident that organizations can have many security policies and even more ideas how to test and control them. Together, they form a so-called Internal Control System (ICS):

> *"An ICS is part of the enterprise management system that ensures the implementation of the enterprise's goals […] and an effective control of work risks, which enables the number of intentional and unintentional mistakes to be minimized" [Lakis12]*

The internal control goal is to ensure compliance, to identify and manage risks, to avoid delays, and to detect defects. An ICS consists of all rules for controlling the compliance of an organization and for reducing the risks of a strategic deviation.

The Security Test Analyst supports any security policy by defining security test approaches. The task of the Security Test Engineer is to execute these test activities. For example, if a security policy is defined to encrypt all data transfers from an application to an outside environment, a Security Test Engineer should apply a dynamic security test to test that an application does not accept unencrypted data.

The number of individual security controls within an ICS can be very high in large organizations. In addition, the frequency of having to apply the controls can also be high. As a result, (test) automation is essential and should be supported by the Security Test Engineer.

3.2.2 Applying security control activities

To support the control activities as part of an ICS, the following activities are essential:

- Defining security test activities.
- Measuring security policy application.

Defining security test activities within an ICS

Given that a security policy is a set of objectives which define what to do in a particular security situation, the Security Test Analyst is tasked with defining security tests which check the application of a security policy.

Establishing a security test control system for a security policy is typically based on an iterative, incremental process, which can be modelled by the Plan, Do, Check, Act (PDCA) cycle as follows:

- Plan: In the plan step, a desired idea of what to do in a particular situation is defined within the context of a security policy. In general, the more specific a security policy is, the easier it is to understand, follow and test it. It is best to define test control aspects in parallel with the security policy.
- Do: In the SDLC, the security policy should be applied by the developers.
- Check: In this step, the test execution takes place. The test result should have only two possible types. A security policy test is said to have passed if there is no deviation between expected and actual results. A security policy test is said to have failed if there is a deviation between expected and actual results.
- Act: In the act step, any defect which violate the security policy should be fixed so that they conform with the security policy.

© International Software Testing Qualifications Board

A false negative occurs when a test fails to detect an existing vulnerability or policy violation. A false positive occurs when a test incorrectly reports a vulnerability or policy violation that does not actually exist.

False positive and false negative results become apparent in the act step. They are a good indicator for the need to refine the security controls and the underlying test procedure. Sometimes, even the security policy itself should be further refined to avoid any anomalies.

Measuring security policy application

If the ICS is well established and the test activities it contains are optimized, the Security Test Engineer is responsible for executing the test activities on a regular basis. The following aspects are important when applying such security control activities:

- The test data, test environment, and test tooling must be well defined to allow for accurate and repeatable test results. Any subjectivity, prejudice or ad-hoc interpretations when applying control activities must be avoided: Repeating a test input must yield the same test result each time.
- Usually, the schedule and frequency of test execution are well defined within the security policy or by performing overarching activities such as creating the financial report for an organization by an external auditor. It is important for the Security Test Engineer to execute the test cases exactly based on these prerequisites. If a test result or control cannot be analyzed due to missing tests, the overall risk analysis may have a blind spot in the security policy. In extreme situations, this might result in an inaccurate and incomplete overall security report.
- It is obvious that reproduceable tests and regular test executions having the same test context are a good indicator for test automation. However, it is fundamental for a security test engineer to do spot checks, i.e., to double-check if the security policy and control activities are still in line with the system under test. If any anomaly is identified, it is very important to adjust the security policy, the system itself, the security test, or the test automation to avoid any anomalies.

3.3 Defining Test Activities from Regulations

Regulatory requirements increasingly define not only what security objectives must be achieved, but also what evidence organizations need to provide. As a result, regulations can be used as a systematic source for deriving security-related test activities.

3.3.1 Using regulations to derive test activities

Generally, the more critical a domain is in which IT is applied; the more regulations govern it. For example, an IT system used to keep an aircraft flying must fulfill more regulations (e.g., by fulfilling specific policies) than an IT system used to manage the entertainment system in an office. The number of regulations has increased over the years: "*Regulation spread to more and more sectors of the economy, and the relative importance of such heavily regulated sectors as transportation, energy, and telecommunications has also increased*" [Fromm81]. Different types of regulations exist. In the security area the focus is on qualitative regulations, which are meant as protective regulations to "*reduce the amount of learning about product quality that takes place during the first few years a product is marketed*" [Fromm81]

Regulations are typically rules that are established, publicized, and maintained by authorities. Sometimes, following a standard is required in a regulation. Sometimes industry standards are reused (and maybe adjusted) for regulation. The security test engineer syllabus has its own chapter for standards and industry best practices in security.

© International Software Testing Qualifications Board

For many sectors, well-known regulations exist which both developers and security testers (security test engineers as well as security test analysts) must follow. The following list explains some very relevant regulations (depending on region, domain, system criticality) for security test analysis. However, please have in mind, that this list is not complete and might be extended by regional, domain specific or technique specific regulations.

- **General Data Protection Regulation (GDPR):** GDPR is a European Union regulation that focuses on the protection of personal data. Organizations handling personal data must provide data security on a very high level. Security testing helps in identifying and mitigating vulnerabilities that could lead to data breaches and exposing sensitive personal data. An accurate test approach here is to perform GDPR penetration testing.
- **Payment Card Industry Data Security Standard (PCI DSS, V4.0)**: PCI DSS [PCI DSS] applies to organizations that handle credit card transactions. Security testing is a critical component of PCI DSS compliance to determine the security of cardholder data. Executing penetration tests is also a good choice to determine that this standard is complied with. An example could be to mimic a real attack to evaluate the network's infrastructure and applications effectively.
- **Health Insurance Portability and Accountability Act (HIPAA):** HIPAA regulations in the United States govern the security and privacy of health information. Security testing is essential for healthcare organizations to protect sensitive patient data.
- **ISO/IEC 27001:** ISO/IEC 27001 is an international standard for information security management systems. It provides a systematic approach to managing sensitive company information. Security testing is part of the risk management process outlined in this standard.
- **National Institute of Standards and Technology (NIST) Framework:** NIST provides cybersecurity frameworks and guidelines that are widely adopted globally. The NIST Cybersecurity Framework (CSF) and NIST Special Publication 800-53R5 provide recommendations for security testing and assessment.
- **Sarbanes-Oxley Act (SOX):** SOX applies to publicly traded companies in the United States. While it primarily focuses on financial reporting, security testing may be relevant to the extent that it validates the safeguards of financial systems and data.
- **Cybersecurity Maturity Model Certification (CMMC):** CMMC is a framework for assessing and enhancing the cybersecurity posture of defense contractors. Security testing is integral to achieving compliance with different CMMC levels.
- **Open Worldwide Application Security Project (OWASP):** While not a regulation, OWASP publishes open-source information and resources on IoT, system software and web application security. The OWASP Top Ten lists, each for a specific domain, are a widely used resource for identifying and addressing the most critical security risks within those domains. In many contexts OWASP is required in projects/contracts as it would be a regulation.

Not fulfilling relevant regulations might have the following negative impact:

- If a legally enforced regulation applies to a specific industry or a specific enterprise, following this regulation is mandatory. Not following it results in a non-compliant product which is not permitted to be either used or sold. The impact of this might be losing the license to do business, paying some penalties or additional controlling activities by the regulative body.
- Risk indication: As regulations usually stem from a group of experts who try to capture their lessons learned from similar situations, it might be risky not to follow their advice.

The task of the Security Test Analyst is to derive test activities from regulations covering IT security. Depending on the nature of a single regulative requirement this can be done in two different ways:

© International Software Testing Qualifications Board

- Regulations which include control activities. Most regulations already contain specific test activities which can be directly re-used by the Security Test Analyst. A typical example comes from the area of identity and access management (IAM). Most regulations require that each permission is the result of a valid approval process which is stored for later control purposes in a non-modifiable database. In addition to this security policy, most regulations require that every 6 to 12 months a so-called reconciliation check must be done. This test double-checks that every IT permission for a specific system is still the result of a valid approvals process. In addition, many regulations demand that a double-check be performed to determine that every user with permissions on a system is still an employee of the organization.
  In such cases, the Security Test Analyst extracts specific test parameters for a given system and specifies a well-defined set of test activities to follow these additional regulations.
- Regulations without control activities: Some regulations do not include additional control activities. For such situations, the Security Test Analyst must define test activities that cover the regulatory requirements. This is usually done by defining several test cases which include expected test results. These are analyzed after test execution as follows:
  - If an actual result equals the expected result, the assumption can be made that the associated regulative requirements have been fulfilled. For example, if a regulative requirement exists that all external facing apps must be penetration tested on a yearly basis and that all significant vulnerabilities are mitigated within 90 days, there should be at least two test objectives. One verifies that all external facing apps have a penetration test report not older than 12 months, and another verifies if all identified critical vulnerabilities are mitigated within 90 days after identification.
  - If an actual result differs from the expected result, this is a strong indication that the regulation has not been fulfilled. Considering the 90-day example above, it would be a regulative violation if the Security Engineer identifies a critical vulnerability in a system which was already identified 120 days before the penetration test.

In general, defining test activities based on regulations does not differ substantially from the controlling activities included in an ICS. However, test activities that verify if a regulation should be handled much more precisely, as any regulative deviation, (such as missing some regulatory aspects), might have a significant impact on an enterprise. An important testing pre-requisite is to know which regulations are valid for a specific enterprise. As some regulations may only apply to certain activities of an organization, it's also important to define what is in scope and out of scope for a particular regulation. Since regulations can change significantly over time, it is advisable to maintain a so-called regulation register which maintains a current description of all relevant regulations that form the test basis for the Security Test Analyst.

© International Software Testing Qualifications Board

4 Defining Test Objectives, Security Test Strategy and Milestones – 135 minutes (K3)

Keywords

test strategy

Domain Specific Keywords

penetration testing

Learning Objectives for Chapter 4:

4.1 Security Test Objectives and Security Test Strategies

STA-4.1.1 (K2) Summarize test objectives and test strategy for security testing

STA-4.1.2 (K3) Apply appropriate test strategies for security testing within a given project scenario

4.2 Milestones for Security Test Activities

STA-4.2.1 (K3) Implement suitable security test activities in SDLC

4.1 Security Test Objectives and Security Test Strategies

A critical prerequisite for effective security risk management is having a clear understanding of any security test objectives and strategies. These elements serve as the basis which allows the Security Test Analysts to align their test activities within the organization's overall risk appetite. Recognizing the purpose, distinctions, and interdependencies between test objectives and test strategies is essential to determine that test activities effectively address identified security risks and support informed risk-based decision-making.

4.1.1 Test objectives and test strategies for security testing

Security test objectives are the specific objectives or outcomes that a security test aims to achieve. They are derived from the security requirements and expectations of the organization, as well as the relevant standards and regulations. Security test objectives can be divided into four main types:

- Compliance objectives: Supporting that the system meets the applicable security policies, laws, and regulations.
- Risk objectives: Identifying and mitigating the potential threats and vulnerabilities that could compromise the system.
- Assurance objectives: Providing a measured level of confidence regarding the system's security posture, demonstrating that specific simulated attacks were unsuccessful under the defined test scope and conditions.
- Improvement objectives: Enhancing the security status and reliability of the system, and implementing best practices.

Security test strategies provide directions which Security Test Analysts should follow in order to achieve the security test objectives. For example, they define the test scope, test approach, test coverage, test techniques, test tools, and resources for designing and executing security tests.

© International Software Testing Qualifications Board

Security test strategies can vary depending on the type, complexity, and maturity of the system, the risk appetite, and budget of the organization. The following test techniques/types and activities can be considered as part of the security test strategy:

- Penetration testing: A type of dynamic application security testing to assess weaknesses and vulnerabilities without causing harm by mimicking an attacker. [ISTQB-Glossary]
- Security risk assessment: The testing and/or evaluation of the management, operational, and technical security controls in an information system to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system. [NIST-Glossary]
- Vulnerability scanning: A type of static analysis that detects vulnerabilities. [ISTQB- Glossary]
- Security audit: An audit evaluating an organization's security processes and infrastructure. [ISTQB-Glossary]
- Security testing based on OWASP Top 10: This is a practical and risk-oriented approach for identifying the most critical vulnerabilities in web applications and APIs. The general OWASP Top 10 defines the most common and impactful security risks, including broken access control, injection attacks, cryptographic failures, insecure configurations, and authentication weaknesses. OWASP-API focuses on strategies and solutions to understand and mitigate the unique vulnerabilities and security risks on APIs (OWASP-API). Both serves as an industry-standard reference for planning and executing security assessments. [OWASP-Top-10 / OWASP-API]

4.1.2 Applying appropriate test strategies for security testing

The ISTQB® Glossary [ISTQB-Glossary] defines a test strategy as how to perform testing in order to reach test objectives under given circumstances. Some test strategies may be better suited to performing security testing than others.
Security testing must be included in the test strategy to ensure the correct strategy is selected. Strategies are often combined to provide the proper coverage for a particular project or phase of a project. For sequential type projects, different test strategies can be applied which include different test levels and/or test types. In agile software development, different test strategies may be applied during iterations.

The chosen test strategy influences the way one can create and manage security test work packages. Main characteristics for the test strategies in the context of security testing are summarized below:

- **Analytical:** Analytical test strategies aim at analyzing the test basis to verify security test conditions. For example, in requirements-based testing, a Security Test Analyst derives security test cases from the security test conditions that are specified by the requirements. Security tests are then implemented to cover test conditions with test cases. The security tests are subsequently executed, often using the priority of the requirement covered by each security test to determine the order in which the tests will be run. Test results are reported in terms of requirements status which may include information about requirements tested and passed, requirements tested and failed, requirements not yet fully tested, and requirements tests which are unable to be executed (i.e., blocked).
- **Model-based:** Model-based test strategies, such as operational profiling, have the goal of helping to define test inputs and conditions to which the system is subjected, and defining how the application subsequently behaves. For example, applying a model-based security test to a web application requires that the Security Test Analyst develops a threat model adapted from present threats and vulnerabilities or based on modeling frameworks like STRIDE [STRIDE], DREAD [DREAD] or PASTA [PASTA]. In the next step, the Security Test

© International Software Testing Qualifications Board

Analyst derives the most likely attack surface and the relevant risks from this model. This forms the basis for creating appropriate security test cases.

- **Methodical:** Methodical test strategies are essential for identifying potential vulnerabilities in a system or application. This can be done by using security standards like ISO/IEC 27001/27002 [ISO27001-27002], OWASP Web Security Testing Guide (WSTG) [OWASP-WSTG], NIST SP 800-53 [NISTSP80053], or simply by using a checklist or a collection of generalized test conditions. For example, in security testing a simple e-commerce website, Security Test Analysts might use a checklist that identifies the attack vectors and vulnerabilities of input fields and select appropriate test cases for penetration testing.
- **Process-compliant or standard-compliant:** Process-compliant test strategies or standard-compliant test strategies can be used for security tests which follow established frameworks and methodologies, and which align with security standards such as ISO/IEC 27001/27002, OWASP Web Security Testing Guide and NIST SP 800-53. Other security standards exist, such as PCI DSS [PCI DSS], which developers must follow if an application or system involves payment card data transactions. The task of security testing is to verify the correct implementation and determine compliance with the PCI DSS standard by performing vulnerability assessments and executing penetration testing.
- **Reactive:** Applying a reactive test strategy the test team waits to design and implement tests until the software is received, reacting to the actual system under test. This strategy is commonly applied when new threats emerge, penetration tests reveal weaknesses, or real-world attacks expose security gaps. Reactive security testing often includes regression testing, vulnerability scanning, penetration testing, and patch verification. Although this approach is effective for quickly responding to discovered threats, it is usually combined with proactive strategies to improve overall security resilience and reduce the risk of future attacks.
- **Consultative:** Using consultative test strategies, the Security Test Analyst relies on the inputs provided by one or more key stakeholders or external security experts to determine security risks to be covered. For example, in outsourced penetration testing for a web-based application, an organization may give the outsourced security test provider a threat model with which to prioritize potential threats and security risks that they want evaluated against their application. The security service provider can then use this threat model as a basis for executing appropriate penetration tests and attack scenarios.
- **Regression-averse:** Regression-averse test strategies can be used when the Security Test Analyst is implementing vulnerability scans of a newly deployed build of a system under test within a continuous integration/continuous delivery (CI/CD) pipeline. For example, a DevOps approach will be extended to a DevSecOps augmentation, when security testing is included in the regression testing that is performed at intervals on the built code. Such vulnerability scans are performed whenever the application has changed, and a new build is created. This facilitates fast feedback concerning newly identified security weaknesses in the current build of the application.
- **Organizational:** An organizational test strategy provides the foundation for structured and risk-based security testing within an enterprise environment. It defines common objectives, testing levels, responsibilities, methods, and quality criteria to ensure consistent security validation across projects. A Security Test Analyst applies this strategy to identify vulnerabilities, evaluate security controls, and verify compliance with organizational and regulatory requirements. Security testing activities are aligned with defined risks, business priorities, and technical standards.

Test strategies which focus on security testing provide the overarching framework and guidance for security test efforts. Security testware uses these test strategies by breaking them down into actionable activities and tasks, so that security testing is effectively planned, executed, and managed throughout the project lifecycle.

© International Software Testing Qualifications Board

There might be an overlap between the different characteristics. E.g., if a security test analyst follows a methodical strategy, she might automatically follow a process compliant strategy.

4.2 Security Test Activities

A crucial element of an effective and reliable security test strategy is the definition and implementation of appropriate security test activities throughout the software development lifecycle (SDLC). These activities support the early identification and remediation of vulnerabilities, thereby reducing security risks, minimizing project delays, and lowering remediation costs in later development phases.

4.2.1 Implementing Suitable Security Test Activities within Agile and Waterfall-Based SDLCs

The most relevant security test activities associated with the different SDLC phases are summarized below (adapted from OWASP Foundation Security Testing [OWASP-SECURITY-TESTING] and OWASP in SDLC [OWAPS-SDLC]):

1. Security Requirements and Planning Activities

The initial security test activities are established during the requirements and planning phase. In agile environments, these activities are integrated into backlog refinement sessions, sprint planning, user story definition, and definition-of-done criteria. In waterfall-based projects, they are typically performed during formal requirements engineering and project planning phases.
Key activities include the identification of:

- security objectives,
- compliance and regulatory requirements,
- business and operational risks,
- security acceptance criteria, and
- initial security testing scope.

Early integration of security requirements ensures that security testing is embedded within the overall test strategy and significantly reduces the likelihood of introducing critical vulnerabilities during later SDLC phases.

2. Threat Modeling and Security Design Review Activities

The second group of activities focuses on threat modeling (e.g., using the OWASP Threat Modeling Cheat Sheet [OWASP-Cheat]) and security architecture reviews. In agile projects, these activities are performed continuously during feature design, epic definition, or prior to implementing user stories. In waterfall-based projects, they are commonly executed during the system and software design phase. Security Test Analysts normally with the support of software architects identify:

- potential threats and attack vectors,
- trust boundaries,
- sensitive data flows,
- architectural weaknesses, and
- misuse and abuse cases (see also OWASP Misuse/Abuse Testing [OWASP-Misuse-Abuse]).

Additionally, mitigation controls and corresponding security test conditions are defined to address identified risks before implementation activities begin.

© International Software Testing Qualifications Board

3. Incremental Security Verification Activities

The next set of activities is conducted during implementation, integration, and system verification phases. In agile projects, security verification is executed continuously within each sprint using automated and manual security testing techniques. In waterfall-oriented approaches, these activities are generally aligned with integration testing and system testing phases.
Typical activities include (refer to [ISTQB-STE]):

- Static Application Security Testing (SAST),
- Software Composition Analysis (SCA),
- Dynamic Application Security Testing (DAST),
- Interactive Application Security Testing (IAST),
- API security testing, and
- secure code reviews.

Continuous execution of these verification activities enables early detection and remediation of vulnerabilities before release preparation. It is important to notice that some of these activities might have more preconditions than others, e.g. for doing a DAST a running system should be available.

4. Pre-Release Security Validation Activities

Critical security validation activities are performed before deployment or production release. In agile delivery models, these activities may occur before major releases, program increments, or deployment windows. In waterfall-based projects, they are usually executed during final system acceptance and release preparation phases.
These activities typically include:

- penetration testing,
- vulnerability assessments,
- configuration and hardening reviews,
- privilege and access-control validation,
- infrastructure security testing, and
- validation of implemented security controls in production-like environments.

The primary objective is to evaluate the overall security posture of the application and determine whether residual risks are acceptable for production release.

5. Operational Security Monitoring and Continuous Assessment Activities

The final group of activities addresses post-deployment and operational security assurance. In agile and DevSecOps-oriented environments, these activities are performed continuously during operations. In waterfall-based environments, they are often executed periodically after deployment.
Typical operational security activities include:

- continuous security monitoring,
- automated vulnerability scanning,
- periodic penetration testing,
- log analysis and security auditing,
- failure detection and validation,
- compliance verification, and
- security patch and configuration management.

© International Software Testing Qualifications Board

These measures support the identification of newly emerging threats, ensure ongoing compliance, and maintain the long-term effectiveness of implemented security controls.

When defining suitable security test activities within the SDLC, it is essential to integrate security systematically and consistently across all development phases. By addressing security requirements, architecture, implementation, validation, deployment, and operational monitoring activities throughout the SDLC, organizations can proactively mitigate risks, reduce remediation costs, and improve the overall resilience of their applications and systems.

A structured and continuous security testing approach not only enhances product security and software quality but also strengthens stakeholder confidence, regulatory compliance, and customer trust. Conducted incrementally, allow developers to identify and resolve defects before they escalate.

When defining suitable security test activities, it is essential to implement them in an accurate way within appropriate SDLC. By addressing security concerns at every phase (i.e., requirements, design, development, pre-deployment, and post-deployment) organization can mitigate risks, reduce costs, and enhance the overall security of their applications. A structured approach to security testing not only safeguards the product but also strengthens trust with users and stakeholders.

© International Software Testing Qualifications Board

5 Specifying Work Packages for the Security Test Analyst – 150 minutes (K3)

Keywords

security testing, testing work package

Learning Objectives for Chapter 5:

5.1 Defining Security Test Work Packages

STA-5.1.1 (K2) Summarize the characteristics of software testing work packages

STA-5.1.2 (K3) Apply specifics for security testing work packages

5.2 Creating a Security Test Plan based on Security Test Work Packages

STA-5.2.1 (K2) Explain the different aspects involved in security test planning

STA-5.2.2 (K3) Implement a test plan for security testing work packages

5.1 Defining Security Test Work Packages

The Security Test Analyst is involved in the planning and organization of the security testing. As discussed in the Security Test Engineer syllabus [ISTQB-STE] the organization and management of security testing can be dependent on various attributes, such as size, complexity, criticality of the system under test, available resources, and the structure of the organization.

5.1.1 Characteristics of software testing work packages

Managing testing assignments and workload involves breaking down the testing project into smaller testing work packages (WPs) and assigning these to different individual testers or groups. The work break down structure (WBS) method is often used in project management [Project Management Institute] (PMI) for creating work packages.

A testing WP can be defined as one or more test-related tasks to be performed by one or more people to realize a specific testing objective. In essence, a testing WP is an assignment which can be communicated orally but should preferably be available in written form as a single-page description or as a User Story.

Typical work products from software testing WPs include test plans, test cases, defect reports, test results, and test reports.

In this syllabus, software testing WPs are defined as assignments to be completed as part of a testing project.

© International Software Testing Qualifications Board

The following may be included in a WP description:

WP Content	Tasks of the Security Test Analyst
Scope	Explicitly define what is included and excluded in the WP.
Objectives	Specify the goals that the WP aims to achieve.
Deliverables	List the tangible outputs or results expected from the WP.
Activities and tasks	Break down the WP into specific activities and tasks.
Resources	Identify the human, technical, and financial resources required to complete the WP.
Schedule	Establish timelines and deadlines for completing the tasks within the WP.
Dependencies	Identify any dependencies between this WP and other tasks or WPs.
Risks and assumptions	Outline potential risks and assumptions related to the WP.
Acceptance criteria	Define the criteria that must be met for the WP to be considered complete.
Compliance	Specify any regulatory or compliance standards that must be adhered to during the testing.
Metrics and reporting	Identify metrics to track progress and methods for reporting status and findings.

5.1.2 Specifics of security test work packages

A number of different aspects may have an influence on defining and completing security test work packages.

Requirements: Security is one of the non-functional quality characteristics described in the [ISO 25010] standard. Information security is one of the most supported quality characteristics by international and national standards, guidelines, and tools. As for other non-functional characteristics, specific and clearly formulated security requirements may be limited or even non-existent. General formulations which state, for example, that an application or system shall be secured against the [OWASP Top 10] vulnerabilities or any other public available security standard are easy to make, but leave many other aspects unspecified.

Time estimates: The estimated time needed for security tests is difficult to predict, and much of the security testing is conducted using an exploratory testing approach. As a result, vulnerabilities could be discovered at any time and may result in a subsequent need for further exploration or verification needing more time. At best, it may be possible to make an estimate for the automated aspects of the security test, but the manual aspects remain difficult to estimate. Prioritizing and time boxing may help to keep control of these uncertainties.

Test environment: A security test may require a specifically designed test environment to enable vulnerabilities to be identified and to assess the software´s resilience against possible attackers. The test environment should be isolated from production systems, and should mirror them as closely as practically possible as it can be. Virtualization platforms such as VirtualBox or VMware are commonly used to create such isolated environments, allowing testers to work within a controlled test environment without risk to live systems. The test environment should be equipped with test tools to conduct different security tests, such as vulnerability scanning and penetration testing. Security testing may potentially damage data, configurations, or even system functionality.

© International Software Testing Qualifications Board

It is therefore essential to take snapshots or backups before testing begins, and to be able to recover and restore these elements to support continuation of the security test.

Criticality: The identification of defects, such as exploitable vulnerabilities, can be critical for a business. Since most businesses rely on their data and other assets for their operations, protecting these assets has the highest priority. Significant management attention and urgency may be applied to resolve these defects as quickly as possible.

Integration: Many companies face challenges integrating security into their design and implementation process. This leads to the majority of security testing being performed towards the end of the project.

Competency: Despite the availability of guidelines and other documentation, security testing requires specific competencies. The configuration of a scanning tool, for example, requires specific know-how. Manually exploring reported issues may require other tooling and competencies to be available.

Work package characteristics: Security test work packages have some additional and unique characteristics. These need consideration to meet the specific requirements and challenges of securing the security of software products. The following are characteristics for security test work packages:

- Focus: This is on identifying vulnerabilities in the software that could be exploited by attackers. A threat analysis may help to identify areas for the security test.
- Scope: This is more targeted on only security testing. It may be assisted by a threat analysis targeting potential security risks, such as user authentication, data encryption, and authorization controls.
- Test types: This may include security-focused test types such as penetration testing (simulating an attacker's behavior), vulnerability scanning (using automated tools to find weaknesses), and security code reviews.
- Tools used: More than in other areas of testing, test tools for security testing require training, competence, and experience in using the tools properly.
- Deliverables: As mentioned in the Software Test Engineer syllabus [ISTQB-STE], security testing provides reports on the details of vulnerabilities found, potential exploits, and recommendations for mitigation. These are expected to be handled differently compared to other test reporting (see chapter 6).

Approach followed: When creating security test work packages for specific software, it is important to consider the objective for this software and the related security risks. This requires a systematic approach to identify and address the specific security requirements and risks associated with the application. The following approach may help when creating security test work packages:

- Start by thoroughly understanding the nature and purpose of the application. Identify its functionality, the type of data it handles, its intended users, and any regulatory or compliance requirements that must be met.
- Perform a risk assessment (see section 3.1).
- Consult relevant security standards, guidelines, and best practices applicable to the type of system under test (see [ISTQB-STE]). This could include, for example, OWASP guidelines, NIST recommendations, or industry-specific standards.
- Consider any regulatory or compliance requirements that apply to the application, such as GDPR, HIPAA, PCI DSS, and others. Determine that the selected security work package aligns with these compliance requirements.

© International Software Testing Qualifications Board

- Assess the different security test techniques and test types available, such as penetration testing, vulnerability scanning, code review, and security architecture review. Determine which test techniques are most suitable for the application's characteristics and security objectives.
- Prioritize the security objectives based on the criticality of the application and the potential impact of security breaches. Focus on addressing the most significant risks first while considering resource constraints and project timelines.
- Customize the security work package to align with the specific security requirements and risks identified for the application. Define the scope, objectives, activities, tasks, resources, and timelines accordingly.
- Seek input from peer security experts or consultants who specialize in information security. They can provide valuable insights and recommendations for selecting the most appropriate security work package and implementing effective security measures.
- Review the selected security work package periodically and refine it as needed to account for any changes in the application's environment, emerging threats, or new vulnerabilities discovered.

5.2 Creating a Security Test Plan based on Security Test Work Packages

Test planning refers to the activity in which the specific steps are determined for how and when to complete a test project or assignment. This includes prioritizing, laying out time schedules, establishing the budget, setting milestones, assessing risks, and solidifying tasks, and assigning them. The Security Test Analyst is expected to design and create a security test plan for the defined security test work packages.

5.2.1 Different aspects involved in security test planning

Test plans can include work packages (WPs), activities, or tasks. Creating a test plan involves outlining project scope, objectives, and timeline so that all stakeholders are on the same page.

Test planning helps to complete testing successfully and avoid potential problems and roadblocks. Benefits of a good test plan include the following:

- Clear test objectives and test scope provide understanding regarding what to achieve and which work products will be delivered.
- Enabling collaboration and communication between stakeholders.
- Identification of potential project related risks and issues.
- Setting clear stakeholder expectations.

The Security Test Analyst needs to consider the following security-specific aspects:

Identifying the test scope of security testing:

- What is in scope and what is out of scope?
- Is the defined scope sufficient for the given test?
- What can be achieved considering the resources and project constraints?

Identifying who should perform the security tests:

© International Software Testing Qualifications Board

- Does the organization have people with appropriate security testing skills and knowledge?
- Does the organization feel comfortable outsourcing the security test?
- In the case of Commercial-Off-the-Shelf software, who is responsible for which security testing?
- Do Security Test Engineers need training in the use of specific security test tools?

Planning an appropriate test schedule:

- Which security-related items need to be implemented and tested before other testing occurs?
- When will security features be available for testing?
- Estimate the security testing schedule.

Defining the activities and tasks to be performed and estimating the time required for each:

- Time needed to design and implement the appropriate security tests?
- Time needed to evaluate and report the test results of security tests?
- Time needed to perform regression tests related to security?
- Time needed to establish the security test environment?

Defining the security test environment(s):

- What is the extent of the environment? (platform, technology, size, location).
- Which security test tools and other test tools must be available?

Procuring authorizations and approvals for security test activities:

- Who needs to authorize and approve security tests?
- When is that authorization needed?
- Are budget and funding sufficient?

5.2.2 Implementing a test plan for security work packages

Security test planning results in a security test plan for the chosen security test strategy. The test plan has to be implemented so that all the various identified work packages are executed.

Note that in this syllabus, the term “Implementation” is defined as the activity to operationalize and execute the test plan.

Work packages are specified as manageable and executable units. The next activity is to schedule selected WPs, acquire the resources needed, agree on matters of content and deliverables, and initiate test monitoring and test control activities.

Test schedule

The most important task is to create a test schedule for the security test. Based on the size of the WPs and available test tools and resources, the Security Test Analyst may need to further break down the WPs so that all prioritized tasks can be executed.

The created schedule allows for monitoring, control, and reporting of progress.

© International Software Testing Qualifications Board

Security test resources

Based on the content of the WP, specific resources, such as people and test tools, may be needed. The Security Test Analyst should know about possible tool options and have knowledge about the possible internal or external human resources with the required competencies.

If the required internal resources are not available, the work will need to be outsourced. During the implementation phase, the Security Test Analyst acquires external resources which are capable of executing the necessary testing.

WP content

When the resources are contacted (and possibly contracted), they and the Security Test Analyst (as the sourcing party) need to agree on what the content of the package is. It is essential that both parties have a common understanding about what must be done.

The testware

There may be one or more work products, depending on the WP size. Documented acceptance criteria for security tests must be defined and approved prior to the commencement of testing, providing a clear and agreed upon basis for evaluating testware. Referring to external public standards and/or guidelines may help to do this.

It must be possible that it is possible to verify and accept the testware when in progress and when it is close to completion.

Test monitoring and Test control

The sourcing party must make clear how test progress and interim quality are monitored. This may result in requesting access to test tools and other resources used, or for interim security test reports to be delivered. The delivering party should specify which possibilities exist for the correction of any deviations from the agreed test plan and test schedule.

All testware, including the security test plan, should be reviewed to assess completeness and correctness. Since security tests are often technical in nature, a technical review session may be the most **appropriate** method. However, walkthroughs and inspections may also be suitable.

© International Software Testing Qualifications Board

6 Interpreting Test Results and Defining Actions – 105 minutes (K4)

Keywords
common vulnerabilities and exposures, common vulnerability scoring system, test report, test result

Learning Objectives for Chapter 6:

6.1 Interpreting Test Results

STA-6.1.1 (K2) Explain the key aspects of test report communication

STA-6.1.2 (K2) Explain management of security test report follow-ups to support timely vulnerability mitigation

6.2 Acting Upon Test Results

STA-6.2.1 (K4) Analyze a given security test report to prioritize necessary actions

6.1 Interpreting Test Results

Security testing does not end with the identification of findings. To create value, test results must be communicated effectively, understood by relevant stakeholders, and translated into appropriate remediation actions. This requires both clear reporting and systematic follow-up activities.

6.1.1 Key aspects of test report communication

Reporting security test results requires close cooperation between the STE and STA. The STE is responsible for creating the security test report, and the STA is responsible for following up and prioritizing any corrective actions needed.

Depending on when the security test is performed, critical security vulnerabilities must be discussed immediately by the STE and STA, and actions taken to manage such emergencies according to defined procedures. If a critical defect is found, it is not advisable to wait until a formal security test report is issued. This is irrespective of whether a test environment or the production system was used.

The content and format of a security test report may vary depending on the stakeholders who are to receive it. Developers and architects need as much detail as possible regarding any vulnerability so they can design and implement the needed fixes. To enable this, the security test reports must contain sufficient details to replicate the security vulnerabilities. These detailed security test reports may also contain possible recommendations provided by the security test tools for each security vulnerability found. Those stakeholders who are responsible for making business decisions need high-level information.

This stakeholder management is a critical responsibility for authors of security test reports. While the technical findings may remain unchanged, their presentation must be tailored to the information needs, responsibilities, and decision-making authority of the intended audience. Executive stakeholders typically require a concise overview of business risks, potential impacts, and recommended actions, whereas technical teams need detailed evidence, attack paths, reproduction steps, and remediation guidance. Effective reporting therefore balances accuracy, transparency, and relevance, ensuring that each stakeholder group receives the level of detail necessary to understand the significance of the findings and to take appropriate action. A well-designed security test report

© International Software Testing Qualifications Board

translates technical vulnerabilities into stakeholder-specific insights, enabling informed decisions and fostering organizational support for risk mitigation activities.

The STA defines the security test report formats, which determine the sections of the report into which specific test results are to be inserted. In addition, the STA must define communication channels and formats in advance of the report being issued, which includes a distribution list of all those who should receive particular security test reports and states how to send them (e.g., only via encrypted email).

Vulnerability scans report on the presence of publicly known vulnerabilities, known exploits or weaknesses. For example, if the software uses a version of an external dependency with a known vulnerability, the software itself might be vulnerable as well. In this case, looking up the corresponding Common Vulnerabilities and Exposures database (CVE) identifier mentioned with the vulnerability helps avoid ambiguity and reduces the risk of misinterpretation [ISTQB-STE]. Vulnerability details are often publicly available, such as those published in the National Vulnerability Database (NVD) [NVD], which can provide further useful context.

6.1.2 Managing security test report follow-ups for vulnerability mitigation

The STA is responsible for communicating security test reports and possible vulnerabilities in a form that enables all necessary mitigation actions (or other risk treatment methods) to be initiated.

The communication for non-emergency-findings starts with discussing the security vulnerabilities with relevant stakeholders once a reviewable security test report has been produced by the STE. The STA's task is to explain the technical details to different stakeholders, including higher management. During this discussion, the STA shall formulate possible follow-up actions and recommendations. These may include temporarily shutting down the system, addressing the security vulnerabilities, or accepting the risks.

Follow-up actions depend on several factors:

- Severity of the vulnerability. These are usually classified into specified categories such as low, medium, and high or are specified using the latest version of the Common Vulnerability Scoring System (CVSS) [CVSS] [ISTQB-STE].
- Required security level and business criticality of the application.
- Possibilities for changing code and the environment.

Apart from the technical details of the vulnerability itself, the severity may also depend on other factors, such as:

- Location of the vulnerability: This could be in a functional component identified as having a high-risk level.
- Defined security requirements: These were defined because they are considered to be important for securely operating the application.

If the vulnerability is linked to a known vulnerability, the publicly available details must be stated, including its severity and exploitability. This helps determine the severity of the vulnerability. Note that the severity of the vulnerability often does not match the vulnerability's severity (e.g. if a system is not connected to Internet at all), but it provides a useful basis for localizing common severity levels and for further refinements (see [ISTQB-STE] and the difference between base CVSS and context-adjusted CVSS).

© International Software Testing Qualifications Board

6.2 Acting Upon Test Results

The primary objective of security testing is not merely to identify weaknesses, but to support decision-making and risk treatment. Therefore, the analysis of test reports and the prioritization of resulting actions constitute essential steps in transforming findings into effective security improvements.

6.2.1 Analyzing security test reports to prioritize necessary actions

Prioritization: This defines the order in which the vulnerabilities should be addressed. Prioritization can be directly linked to the severity of a vulnerability: For example, a vulnerability with a CVSS score of 10.0 (the highest score) must usually be fixed as soon as possible and thus warrants performing an immediate release update in production. An exemption might be a contextual adjustment (e.g. an exploitable system might not be connected with internet at all). By comparison, a vulnerability with a CVSS score of 1.0 can be fixed and deployed as part of the next planned releases. Note that business objectives may also influence prioritization levels. In general, the following summary of high-level steps can serve as a guide for the prioritization:

STA Task	STE Task	Task description
x		Define security test work packages.
x		Define stakeholder and audience-specific security test reports, formats, and communication channels.
	x	Execute security test work packages and generate security test reports.
x	x	Create a suggestion for prioritizing identified vulnerabilities.
x		Communicate the security test report and prioritization.
	x	Perform confirmation tests for the implemented fixes.

Severity: This is a major factor in deciding the prioritization, but should not be the only factor considered in that decision. Depending on the context and the domain, factors may apply that are equally as important, for example:

- Some vulnerabilities may have a high severity but are design defects that require re-design and/or rewriting major parts of an application to mitigate them. In such cases, it may not be reasonable to assign them a high priority. Mitigations often involve multiple steps. A common approach is to first take high-priority measures to lower the risk and then implement a complete fix later with lower priority.
- Low-severity vulnerabilities might be in critical functional areas and show signs of active exploitation. They may also represent a high risk of damage or loss if exploited. These factors can increase the priority of the vulnerability.
- In cases where a number of vulnerabilities take a long time to resolve, it could be advisable to prioritize the defects that are deemed to be quick to resolve.
- When vulnerabilities exist that have not been resolved for some time, it might be advisable to give them a higher priority because the likelihood that they are already known to attackers is higher. Such vulnerabilities have been proven to be present for a long time, in contrast to the vulnerabilities that were only recently introduced.
- Sometimes multiple vulnerabilities can be fixed by making changes in the same or a similar area, or where the recommended actions overlap. In such cases, it is advisable to prioritize

© International Software Testing Qualifications Board

them closely together. Fixing them together is likely to be faster than changing the same area multiple times separately, as they can be considered as a whole and fixed in one change.

- Vulnerabilities that are obviously visible but have a low severity could be prioritized higher if they give a bad impression of the application's security posture or lead to a compound risk. For example, exposing the version numbering dependencies used in test logs that are shown to the user could have a low severity if the versions are regularly updated. However, it could give the impression to attackers that the application is an easy target if it doesn't follow basic security best practices, or the version number could be used to find potential vulnerabilities.
- Even if a security test report already contains recommendations for necessary mitigation actions, they should not be followed blindly and considered as the only actions to perform. They may not cover aspects that go beyond mitigating the specific defect, and which could be equally important to consider improving overall security.

Risk levels: Actions taken to counteract vulnerabilities can be applied to the processes, organization, employees, or work products/deliverables. To define and propose appropriate actions for a vulnerability, it is first necessary to perform a risk analysis. After all, defining actions that involve a lot of effort to resolve a vulnerability whose risk is very low might not be appropriate, for example:

- If a vulnerability cannot be mitigated by fixing the code, the risk level could be reduced by lowering the impact of the vulnerability being exploited (e.g. by not using the affected system any more for critical use cases).
- If the vulnerability relates to potential unauthorized physical access to servers, the risk level could be transferred or shared by either outsourcing the server hosting to a different organization or by insuring against the damage should the event occur.
- If secure storage of Personally Identifiable Information (PII) has a high-risk level, the risk can be avoided by not storing any PII in the first place. However, this is not always an option and may mean losing potential gains that could have resulted from accepting the risk.

Extent of actions: Actions may not be limited to the immediate treatment after identifying the risk level of a vulnerability, but may also extend beyond that. For example, it is not enough to perform security awareness training with employees once, it is necessary for new employees to get trained as well as retraining employees at regular intervals. In addition, if SQL injections are found regularly, for example, it is not enough to mitigate the vulnerabilities by fixing them as they are found. Measures should be taken to avoid SQL injection vulnerabilities are not introduced in the first place, for example, by regularly training developers in secure coding or establishing a code review process.

Combining actions: Actions may not directly target a single individual vulnerability. It could be more effective to define actions that consider similar or related vulnerabilities. If the majority of vulnerabilities are related to vulnerabilities in the code, considering each vulnerability on its own would only result in the vulnerabilities being addressed directly. Having a large number of such vulnerabilities may be an indicator of poor knowledge of secure coding practices by the developers, which makes training them a necessary additional action to support long-term improvements.

When determining appropriate actions, it is important to consider past actions taken against similar vulnerabilities. For example, if a vulnerability keeps re-occurring, this indicates that the actions taken are not effective enough or they were not executed properly. This makes it necessary to define different actions from those tried previously or to make changes to previously defined actions.

Scope: The scope of an action should be considered. For example, if a certain group of employees does not have any access to IT systems, it might not be beneficial to train them in security awareness at all. On the other hand, it might make sense to make the retraining interval of security awareness training shorter for employees with access to very critical infrastructure compared to employees with access to less critical infrastructure. Other actions, however, make more sense when performed globally, such as installing anti-virus software, as it is usually cheaper to have the same setup for every employee.

© International Software Testing Qualifications Board

7 References

Standards

[ISO 25010] ISO/IEC 25010:2023 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Product quality model

[ISO 27000] ISO/IEC 27000 family - Information Security Management

[ISO 27005] ISO/IEC 27000 family - Information technology — Security techniques — Information security risk management

[ISO 31000] ISO 31000:2018: Risk management – Guidelines

[PCI DSS] Payment Card Industry Data Standard Security (PCI-DSS)

ISTQB® Documents

[ISTQB-ALTA] ISTQB® Advanced Level Test Analyst Syllabus, Version 4.0

[ISTQB-ALTM] ISTQB® Advanced Level Test Management Syllabus, Version 3.0

[ISTQB-ALTTA] ISTQB® Advanced Level Technical Test Analyst Syllabus, Version 4.0

[ISTQB-CTFL] ISTQB® Foundation Level Certified Tester Syllabus, Version 4.0

[ISTQB-ELTM-SM] ISTQB® Expert Level Test Management – Strategic Syllabus, Version 1.0

[ISTQB-STE] ISTQB® Security Test Engineer Syllabus, Version 1.0

Glossary References

[Camb23] Cambridge Dictionary, under “Policy”, online available at https://dictionary.cambridge.org/dictionary/english/policy

[ISTQB-Glossary] ISTQB Glossary, online available at https://glossary.istqb.org

[NIST-Glossary] Computer Security Resource Center: Glossary, actual version, online available at https://csrc.nist.gov/glossary

Books, Articles and Web Pages

[AI26] IVA: AI-driven Cyber Threats The Next Twelve Months, online available at : https://www.ncsc.se/siteassets/publikationer/ai-driven-cyber-threats-iva.pdf

[Alberts03] Alberts, C., Dorofee, A., Stevens, J., & Woody, C. (2003): "Introduction to the OCTAVE Approach", Software Engineering Institute, Carnegie Mellon University, online available at: https://www.sei.cmu.edu/library/introduction-to-the-octave-approach/

[BSI24] Bundesamt für Informationssicherheit: “Social Engineering - the "Human Factor", online available at https://www.bsi.bund.de/EN/Themen/Verbraucherinnen-und-Verbraucher/Cyber-Sicherheitslage/Methoden-der-Cyber-Kriminalitaet/Social-Engineering/social-engineering_node.html

[CISA] The Cybersecurity and Infrastructure Security Agency (CISA): Defining Insider Threats

[Cora2023] Sofia Cora: “Shoulder Surfing: What is it and how to Stop it”, in Fordham Security IT, Information Security New, 24th October 2023, online available at https://itsecurity.blog.fordham.edu/2023/10/24/shoulder-surfing-what-is-it-and-how-to-stop-it/

[CVE] Common Vulnerabilities and Exposures at: https://cve.org

© International Software Testing Qualifications Board

[CVSS] Common Vulnerability Scoring System at https://first.org/cvss/

[Cyber] Cybersecurity Insiders, online available at "Motivations for Hackers to launch Cyber Attacks"

[DREAD] DREAD Threat Modeling: https://threat-modeling.com/dread-threat-modeling

[EveSec] Eventus Security,18 th June 2026, online available at: What Is Social Engineering? Types, Examples & Prevention

[FAIR™] FAIR Institute. (n.d.). Factor Analysis of Information Risk (FAIR™). Model Standard Artifact | Version 3.0 January 15, 2025 online available at https://www.fairinstitute.org/hubfs/Standards%20Artifacts/Factor%20Analysis%20of%20Information%20Risk%20(FAIR)%20Standard%20v3.0%20(January%202025).pdf

Fromm81] Gary Fromm: “Studies in public regulations”, MIT Press, 1981

[Gart23] Gartner Glossary “Business Impact Analysis (BIA)”, online available via https://www.gartner.com/en/information-technology/glossary/bia-business-impact-analysis

[Lakis12] Vaclovas Lakis, Lukas Giriūnas: “The concept of internal control system: theoretical aspect”, ISSN 1392-1258. ekonomika 2012 Vol. 91(2), online available at https://epublications.vu.lt/object/elaba:4594571/

[NIST SP80053] NIST Special Publication, Rev. 5 at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf

[NIST 800-37] NIST Special Publication 800-37 Revision 2: " Risk Management Framework for Information Systems and Organizations", online available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r2.pdf

[NIST 800-300] Guide for Conducting Risk Assessments, online available at https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf

[MITRE] Guide for MITRE Attack, online available at https://attack.mitre.org

[Morpus22] Nicholas Morpus: “What are the Methods and Motives for Attacking?”, VMware Security Blog, 26th July 2022, online available at What are the Methods and Motives for Hacking? - VMware Security Blog - VMware

[NVD] National Institute of Standards and Technology at https://nvd.nist.gov

[OWASP-API] Definition of API security testing: https://owasp.org/www-community/api_security_tools

[OWASP-Cheat]: OWASP Threat Modeling Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

[OWASP-Misuse-Abuse]: https://owaspsamm.org/model/verification/requirements-driven-testing/stream-b

[OWASP-Top-10] The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications: https://owasp.org/Top10/2025/ or https://owasp.org/www-project-top-ten/

[OWASP-SDL] A detailed description how to implement OWASP artifacts in SDLC can be found at https://owasp.org/www-project-integration-standards/writeups/owasp_in_sdlc/?utm_source=chatgpt.com

[OWASP-SECURITY-TESTING] A recommended guide for security testing is provided on the OWASP Security Culture web page at https://owasp.org/www-project-security-culture/v11/7-Security_Testing/?utm_source=chatgpt.com

[OWASP-WSTG[OSINT] What is OSINT: Open-source intelligence? Available at: OSINT: Open-source intelligence, data.europa.eu

© International Software Testing Qualifications Board

[OWASP] OWASP Worldwide Security Testing Guide, online available at https://owasp.org/www-project-web-security-testing-guide/ and https://owasp.org/www-project-integration-standards/writeups/owasp_in_sdlc/?utm_source=chatgpt.com

[PASTA] PASTA Threat Modeling: https://threat-modeling.com/pasta-threat-modeling

[Project Management Institute] https://www.pmi.org

[Poremba24] Sue Poremba: “What are insider threats? Definition, types, and how to mitigate them”, online available at What are Insider Threats? Definition, Types and Mitigation | Verizon Business

STRIDE] The STRIDE Threat Model: https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)

[Verizon24] Verizon Business: “Data Breach Investigations Report”, online available at https://www.verizon.com/business/resources/reports/dbir/

[Wright21] Gavin Wright: “Dumpster diving”, 6th of April 2021, online available at What is dumpster diving?

[XYGENI.IO] What Are the Phases of Software Development Life Cycle?, online available at https://xygeni.io/blog/what-are-the-phases-of-software-development-life-cycle/

[VerSprite]: PASTA Threat Modeling: https://versprite.com/blog/what-is-pasta-threat-modeling/

[LINDDUN]: Wuyts, K. et al. LINDDUN Privacy Threat Modeling Framework: https://linddun.org

© International Software Testing Qualifications Board

8 Appendix A – Learning Objectives/Cognitive Level of Knowledge

The specific learning objectives applying to this syllabus are shown at the beginning of each chapter. Each topic in the syllabus will be examined according to the learning objective for it.

The learning objectives begin with an action verb corresponding to its cognitive level of knowledge as listed below.

Level 1: Remember (K1)

The candidate will remember, recognize and recall a term or concept.

Action verbs: Recall, recognize.

Examples
Recall the concepts of the test pyramid.
Recognize the typical objectives of testing.

Level 2: Understand (K2)

The candidate can select the reasons or explanations for statements related to the topic, and can summarize, compare, classify and give examples for the testing concept.

Action verbs: Classify, compare, differentiate, distinguish, explain, give examples, interpret, summarize

Examples	Notes
Classify test tools according to their purpose and the test activities they support.	
Compare the different test levels.	Can be used to look for similarities, differences or both.
Differentiate testing from debugging.	Looks for differences between concepts.
Distinguish between project and product risks.	Allows two (or more) concepts to be separately classified.
Explain the impact of context on the test process.	
Give examples of why testing is necessary.	
Infer the root cause of defects from a given profile of failures.	
Summarize the activities of the work product review process.	

Level 3: Apply (K3)

The candidate can carry out a procedure when confronted with a familiar task or select the correct

© International Software Testing Qualifications Board

procedure and apply it to a given context.

Action verbs: Apply, implement, prepare, use

Examples	Notes
Apply boundary value analysis to derive test cases from given requirements.	Should refer to a procedure / technique / process etc.
Implement metrics collection methods to support technical and management requirements.	
Prepare installability tests for mobile apps.	
Use traceability to monitor test progress for completeness and consistency with the test objectives, test strategy, and test plan.	Could be used in a LO that wants the candidate to be able to use a technique or procedure. Similar to 'apply'.

Level 4: Analyze (K4)

The candidate can separate information related to a procedure or technique into its constituent parts for better understanding and can distinguish between facts and inferences. A typical application is to analyze a document, software or project situation and propose appropriate actions to solve a problem or task.

Action verbs: Analyze, deconstruct, outline, prioritize, select.

Examples	Notes
Analyze a given project situation to determine which black-box or experience-based test techniques should be applied to achieve specific goals.	Examinable only in combination with a measurable goal of the analysis. Should be of form 'Analyze xxxx to xxxx' (or similar).
Prioritize test cases in a given test suite for execution based on the related product risks.	
Select the appropriate test levels and test types to verify a given set of requirements.	Needed where the selection requires analysis.

Reference

(For the cognitive levels of learning objectives)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and

Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

© International Software Testing Qualifications Board

9 Appendix B – Business Outcomes traceability matrix with Learning Objectives

This section lists the traceability between the Business Outcomes and the Learning Objectives of Security Test Analyst

Business Outcomes: Security Test Analyst		Number of LOs			
ID	Description		K2	K3	K4
STA-BO1	Explain assets and their related security risks	4	3	1	
STA-BO2	Understand attacker mindset	4	3	1	
STA-BO3	Understand relations between risk Management, control systems and regulations	5	1	3	1
STA-BO4	Define test objectives, security test strategy and activities	3	1	2	
STA-BO5	Specify work packages for the Security Test Engineer	4	2	2	
STA-BO6	Interpret test results and define actions	3	2		1

6

			Business Outcomes					
LO Number	Learning Objective	K-Level	STA-BO1	STA-BO2	STA-BO3	STA-BO4	STA-BO5	STA-BO6
STA-1.1.1	Explain what an asset is	K2	x					
STA-1.1.2	Explain the value of an asset	K2	x					
STA-1.1.3	Apply risk identification techniques to determine potential security risks to a protected asset	K3	x					
STA-1.2.1	Interpret risk assessment for assets	K2	x					
STA-2.1.1	Give examples of how knowledge of an attacker's motivations can enhance the work of a Security Test Analyst	K2		x				
STA-2.1.2	Apply methods an attacker might use to discover key information that can be leveraged in security testing	K3		x				
STA-2.2.1	Give examples of specific risks posed by an insider	K2		x				
STA-2.2.2	Distinguish between external attackers and insiders in the context of security testing	K2		x				
STA-3.1.1	Explain risk management methodology	K2			x			
STA-3.1.2	Select a risk management methodology for identified security risks	K4			x			
STA-3.2.1	Apply the relation between policies and internal control systems	K3			x			
STA-3.2.2	Apply security control activities	K3			x			

			Business Outcomes					
LO Number	Learning Objective	K-Level	STA-BO1	STA-BO2	STA-BO3	STA-BO4	STA-BO5	STA-BO6
STA-3.3.1	Use security regulations to derive test activities	K3			x			
STA-4.1.1	Summarize test objectives and test strategy for security test	K2				x		
STA-4.1.2	Apply appropriate test strategies for security testing within a given project scenario	K3				x		
STA-4.2.1	Implement suitable security test activities in SDLC	K3				x		
STA-5.1.1	Summarize the characteristics of software testing work packages	K2					x	
STA-5.1.2	Apply specifics for security testing work packages	K3					x	
STA-5.2.1	Explain the different aspects involved in security test planning	K2					x	
STA-5.2.2	Implement a test plan for security testing work packages	K3					x	
STA-6.1.1	Explain the key aspects of test report communication	K2						x
STA-6.1.2	Explain management of security test report follow-up to ensure timely vulnerability mitigation	K2						x
STA-6.2.1	Analyze a given security test report to prioritize necessary actions	K4						x

7

10 Appendix C – Security Test Domain Specific Terms

Term Name	Definition
asset	Description of any data, device, system, or component that supports information-related activities and has value to the organization requiring protection. Assets are items of value to achieve organizational mission/business objectives [NIST-Glossary].
asset inventory	A comprehensive catalogue of all physical and digital assets owned or used by an organization.
attack surface	The set of points on the boundary of a system, a system element, or an environment where an attacker can try to enter, cause an effect on, or extract data from [NIST-Glossary].
business impact analysis	A systematic process used to evaluate the potential effects of disruptions to critical business operations, helping organizations prepare for and recover from unexpected events.
incident	An incident is an unplanned, unexpected, or unwanted event that disrupts normal operations causes damage or poses a risk to people, property or systems.
incident response	The mitigation of violations of security policies and recommended practices [NIST-Glossary].
internal control system (ICS)	A structured framework of policies, procedures, and mechanisms within an organization designed to support operational efficiency, accurate reporting, and compliance with laws and regulations.
penetration testing	A type of dynamic application security testing to assess weaknesses and vulnerabilities without causing harm by mimicking an attacker.
work package	The Work Package is at the lowest level of project planning, control and reporting. The work package defines a clearly defined element of work to be performed that can be assigned to a responsible individual to do or manage [Project Management Institute].

© International Software Testing Qualifications Board

11 Appendix D – Trademarks

ISTQB® is a registered trademark of International Software Testing Qualifications Board

© International Software Testing Qualifications Board

12 Index

All testing terms are defined in the ISTQB® Glossary [ISTQB-Glossary].

analytical test strategies
    security test 33
anomalies..... 45
assets
    asset 11; 12; 13; 14; 15; 16; 23; 24; 27; 54; 56
attack vectors
    attack vector 34
attackers
    attacker..... 18; 40; 54
attacks
    attack..... 22; 32
business impact analysis 16; 23; 56
Commercial Off The Shelf..... 42
compliance requirements..... 40
compliance standards
    compliance standard..... 39
consultative test strategies
    consultative test strategy 34
CVE 45
CVSS 45
exploitable vulnerabilities 40
insiders
    insider 50
intangible assets 11
internal attacker 18
internal control system
    internal control system
        ICS 23; 28; 49; 56
internal control systems
    internal control system 27
        ICS 23
ISO 31000..... 23; 24; 26
ISO/IEC 27001/27002 34
methodical test strategies
    methodical test strategy 34
model-based security tests
    model-based security test 33
model-based test strategies
    model-based test strategy..... 33
NIST SP 800-53..... 34
operational profiling..... 33
OWASP..... 39
    OWASP
        OWASP Testing Guide 34
PCI DSS..... 30; 40
    PCI DSS
        standard 34
penetration testing..... 33
process-compliant test strategies
    process-compliant test strategy 34
regression-averse test strategies
    regression-averse test strategy..... 34
regulation register 31
risk analysis..... 26
risk assessment 16; 17; 23; 24; 26; 27; 40
risk evaluation 26
Risk identification 26
risk management13; 17; 23; 24; 25; 26; 30; 54

© International Software Testing Qualifications Board

security risk management 32
risk treatment 24; 27
safeguards
safeguard 30; 37
security assessment 33
security audit 33
security awareness 20; 47
security events
security event 15
security objectives
security objective 41
security policies
security policy 27; 28; 32
security policy 27; 28; 29; 30
security requirements
security requirement 32; 40; 41; 45
security risks
security risk 14; 23; 25; 26; 32; 34; 40
security standards
security standard 34; 40
security test activities
security test activity 28; 32; 42; 55
security test cases
security test case 33; 34
security test conditions
security test condition 33
security test objectives
security test objective 32
security test report 44
security test reports 40
security test strategies
security test strategy 14; 26; 27; 32; 33; 34; 42; 55
security tests
security test 7; 9; 10; 18; 19; 20; 21; 26; 27; 28; 29; 30; 32; 33; 34; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 55
social engineering 15
software development lifecycle
software development lifecycle
SDLC 28; 37
SQL injections 47
standard-compliant test strategies
standard-compliant test strategy 34
test environment 39
Test planning 41
test results 8; 44; 45; 53
threat model 33; 34
threats
threat 15; 16; 17; 32; 33; 34; 41; 50
time boxing 39
user story 38
vulnerabilities
vulnerability 13; 16; 26; 30; 31; 32; 33; 34; 39; 40; 41; 45; 46; 47
vulnerability scan 33
vulnerability scanning 38; 40; 41
weaknesses
weakness 16; 26; 34; 40; 45
work break down structure (WBS) 38

© International Software Testing Qualifications Board