**Task Description – Reading Technique**

Together with this task description, you received the following documents:

* **Document 1:** A correct functional specification with user stories of a system to be developed and its security specifications that contain defects.
* **Document 2:** Reading technique for detecting defects.

You are asked to inspect the security specifications (see document 1). The goal is to identify the following types of defects:

**DEFECTS TYPES**

**Omission:** Necessary information has been omitted in the security specification.

**Ambiguity:** The security specification has multiple interpretations.

**Inconsistency:** Two or more security specifications are in conflict with one another.

**Incorrect fact**: The security specification is not true under the conditions specified for the system.

Please follow the steps below.

1. When you start the review, please write down the time in document 2.
2. Read and analyze a user story and its security specifications (see document 1).
3. Go to document 2 and consider the high-level security requirements (HLSR column) related to the security property associated to the user story.
4. To detect omission defects, consider the following statement and answer the question:
   * **When comparing the security specifications (see document 1) with the high-level security requirements, are there high-level security requirements or characteristics that were not specified?**
     1. If the answer is affirmative, record the defect in document 2 by marking the column [ Omission] row [ high-level security requirement not specified ] as an “X”.
   * **Look at the “AND” logical connectors**

b) If any of the conditions of the sentence (HLSR) is not met, record the defect in document 2 by marking the column [ Omission ] row [ high-level security requirement partially specified ] as an “X”.

1. To detect ambiguity defects, answer the following question:

* **Does any security specification allow for multiple interpretations?**

1. If the answer is affirmative, record the defect in document 2 by adding to the column [ Ambiguity ] the identifier of the defective security specification.
2. To detect inconsistency defects, answer the following question:
   * **Are there two or more security specifications in conflict with one another?**
3. If the answer is affirmative, record the defect in document 2 by adding to the column [ Inconsistency ] the identifier of the defective security specification.
4. To detect incorrect defects, answer the following question:
   * **Is there any security specification stating information that is not true under the conditions specified for the system?**
5. If the answer is affirmative, record in document 2 the defect by adding to the column [ Incorrect fact ] the identifier of the defective security specification.
6. Repeat the same workflow (step 2, … , step 7) for the next user story.

1. When you finish the review, write down the time in document 2.

Thank you very much for your participation!

**DOCUMENT 1**

**A correct functional specification with user stories and its security specifications.**

|  |  |
| --- | --- |
| **User Story 1** | **Security Specification** |
| As a client, I want to send my personal information so that other systems can use my information. | **1.1** The system shall ensure that there is no residual data exposed. |
| **1.2** The system shall store credentials securely using the AES encryption algorithm. |
| **1.3** The system shall use the RSA encryption algorithm to protect all data all the time. |
| **1.4** The system shall inactivate a session when it exceeds certain periods of inactivity. |
| **1.5** The system shall encrypt the roles and privileges of the system. |

|  |  |
| --- | --- |
| **User Story 2** | **Security Specification** |
| As a admin, I want to backup data and users transactions so I can keep the information available**.** | **2.1** The system shall ensure that logs are not forever, but have an absolute lifetime that is as short as possible. |
| **2.2** The system shall be able to recover the database at any time. |
| **2.3** The system shall store all transactions of the users. |
| **2.4** The system shall backup of all data periodically. |
| **2.5** The system shall backup of all denials of services. |

**DOCUMENT 2**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Defects Notification Form** Review Start Time \_\_\_\_\_\_\_\_\_ Review End Time \_\_\_\_\_\_\_\_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story** | **Security Property** | **OWASP High-level Security Requirements (HLSR)** | **Omission** | **Ambiguous** | **Inconsistent** | **Incorrect Fact** |
| As a client, I want to SEND my personal information so that other systems can use my information. | **Confidentiality (C)** | C1. Sensitive data shall be protected from unauthorized observation or disclosure both in transit AND when stored. |  |  |  |  |
| C2. System sessions shall be unique to each individual AND cannot be shared. |  |
| C3. System sessions are invalidated when timing out during periods of inactivity. |  |
| C4. TLS protocol shall be used where sensitive data is transmitted. |  |
| C5. System shall use strong algorithms and ciphers at all times. |  |
| As a admin, I want to backup data and users transactions so I can keep the information available**.** | **Availability (A)** | A1 The application server shall be suitably hardened from a default configuration. |  |  |  |  |
| A2 HTTP responses contain a safe character set in the content type header. |  |
| A3. Backups must be implemented AND recovery strategies must be considered. |  |

**List of Seeded Defects READING TECHNIQUE**

|  |  |  |
| --- | --- | --- |
| **Security Specification (SS)** | **Defect Type** | **Description** |
| **1.1** | **Ambiguity** | It Is not clear which data should be protected |
| **1.2** | **Inconsistency** | The SS 1.2 and 1.3 are in conflict. The SS 1.3 indicates encrypt all data using the RSA algorithm. Nevertheless, SS 1.2 indicates to protect credentials, which are also data, using the AES algorithm. |
| **1.3** | **Inconsistency** | The SS 1.2 and 1.3 are in conflict. The SS 1.3 indicates encrypt all data using the RSA algorithm. Nevertheless, SS 1.2 indicates to protect credentials, which are also data, using the AES algorithm. |
| **1.4** | **Ambiguity** | “Certain periods of inactivity” is ambiguous because it is not clear the amount of time to execute the control. |
| **1.5** | **Incorrect fact** | Concepts/functionalities of the system cannot be encrypted. The concept “encrypt” is not correct in the statement. |
| **2.1** | **Inconsistency** | The SS 2.1 and 2.3 are in conflict. The SS 2.1 states that logs are not forever. On the other hand, the SS 2.3 states that all transactions must be stored. This apparently indicates that logs are always generated and stored |
| **Ambiguity** | It not clear what is a transaction. Is it the user data? |
| **2.2** | **------** | **OK** |
| **2.3** | **Inconsistency** | The SS 2.1 and 2.3 are in conflict. The SS 2.1 states that logs are not forever. On the other hand, the SS 2.3 states that all transactions must be stored. This apparently indicates that logs are always generated and stored |
| **2.4** | **Ambiguity** | “Periodically” is an ambiguous word. A time interval is not defined. |
| **2.5** | **Incorrect fact** | Denial of service is an attack, hence is not possible to backup this. |

|  |  |  |  |
| --- | --- | --- | --- |
| **HLSR** | **User Story** | **Defect Type** | **Description** |
| C2 | 1 | Omission | The high-level security requirement is not defined in the specification. |
| C4 | 1 | Omission | The protocol to transport the data is not defined. Note that the protocol define the algorithm to be used in the communication. |
| D1 | 2 | Omission | The high-level security requirement is not defined in the specification. |
| D2 | 2 | Omission | The high-level security requirement is not defined in the specification. |