**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 admin, I want to DELETE the personal information of customers so I can keep depured my database. | **1.1** The system shall verify all SQL queries and thus not susceptible to SQL injection. |
| **1.2** The system shall identify the originator of any information before that information is used by the system. |
| **1.3** The system shall protect sensitive data from external attacks using controls such as a host-based firewall or OS patching. |
| **1.4** The system shall implement data validators such as whitelisting, greylisting or blacklisting). |
| **1.5** The system shall ensure that all users have admin permission (to read, write, delete). |

|  |  |
| --- | --- |
| **User Story 2** | **Security Specification** |
| As a user, I want to LOGIN so I can access to certain information. | **2.1** The system shall prevent that the password from being displayed in clear text. |
| **2.2** Failed login attempts shall record for audit trail and incident reporting purposes. |
| **2.3** The system shall encrypt the credentials stored in the database using TLS protocol. |
| **2.4** The system shall only accept complex passwords. |
| **2.5** The system shall restore the user's password for each failed attempt. |

**DOCUMENT 2**

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

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story** | **Security Property** | **High-level Security Requirements (HLSR)** | **Omission** | **Ambiguous** | **Inconsistent** | **Incorrect** |
| As a admin, I want to DELETE the personal information of clients so I can keep depured my database. | **Integrity (IN)** | I1. Any unauthorized modification of data must yield an auditable security-related event. |  |  |  |  |
| I2. All input is validated to be correct and fit for the intended purpose (e.g., queries parameters, strings variables, REST calls and cookies). |  |  |  |  |
| I3. Data from an external entity or client shall never be trusted and shall be handled accordingly. |  |  |  |  |
| As a user, I want to LOGIN so I can access to certain information. | **Identification & Authentication (IA)** | IA1. Users are associated with a well-defined set of roles and privileges. |  |  |  |  |
| IA2. The digital identity of the sender of a communication must be verified. |  |  |  |  |
| IA3. Only those authorized are able to authenticate and credentials are transported AND stored in a secure manner. |  |  |  |  |
| IA4. Passwords treatment must include complex passphrases, options to recover and reset the password AND default passwords not allowed. |  |  |  |  |

**List of Seeded Defects**

|  |  |  |
| --- | --- | --- |
| **Security Specification (SS)** | **Defect Type** | **Description** |
| **1.1** | **Omission** | According to the HLSR I2, all input must be validated, not only SQL queries. Thus, this SS has a defect. |
| **1.2** | **------** | **OK** |
| **1.3** | **Ambiguity** | The SS reflects a weak statement, “such as” indicates an unclear definition of the control to be implemented. |
| **1.4** | **Ambiguity** | The SS reflects a weak statement, “such as” indicates an unclear definition of the control to be implemented. |
| **1.5** | **Incorrect** | All users cannot have admin permissions. |
| **4.1** | **------** | **OK** |
| **4.2** | **------** | **OK** |
| **4.3** | **Omission** | According to HLSR IA3, credentials are transported and stored in a secure manner. However, the SS states only encrypting credentials at rest. |
| **Incorrect** | TLS is not an algorithm. It is a protocol that defines the algorithm to be used. |
| **4.4** | **Omission** | According to HLSR IA4, passwords should not only consider complex password. There are others items to be consider. |
| **Ambiguous** | Complex passwords is ambiguous. Is it a password with capital letter only and special characters? Is it a password with 10 characters length?. |
| **4.5** | **Incorrect** | Restoring the user's password for each failed attempt would not be correct for the user, therefore, this is incorrect. |

|  |  |  |  |
| --- | --- | --- | --- |
| **HLSR** | **User Story** | **Defect Type** | **Description** |
| I1 | 1 | Omission | The high-level security requirement is not defined in the specification. |
| IA1/C6 | 4 | Omission | The high-level security requirement is not defined in the specification. |
| IA2/C7 | 4 | Omission | The high-level security requirement is not defined in the specification. |