Final Security Requirements Report

Mobile Platform	Android App ; IoT System
Application domain type	m-Health
Authentication	Yes
Authentication schemes	Biometric-based authentication ; Factors-based authentication
Has DB	Yes
Type of database	SQL (Relational Database)
Which DB	MySQL
Type of information handled	Personal Information ; Confidential Data ; Critical Data
Storage Location	Both
User Registration	Yes
Type of Registration	The users will register themselves
Programming Languages	Java
Input Forms	Yes
Upload Files	Yes
The system has logs	Yes
The system has regular updates	Yes
The system has third-party	Yes
System Cloud Environments	Hybrid Cloud
Hardware Specification	Yes
HW Authentication	Basic Authentication (user/pass)
HW Wireless Tech	3G ; 4G/LTE ; 5G ; Bluetooth ; Wi-Fi ; GPS ; RFID ; NFC
Device or Data Center Physical Access	Yes

Confidentiality

Confidentiality Requirement in Security Engineering

Confidentiality is a critical security requirement in the engineering of any system. It ensures that only authorized users can access sensitive information, protecting the system from unwanted disclosure of information. Confidentiality can be achieved through the implementation of encryption, authentication, and access control mechanisms. Additionally, physical security measures such as restricted access to areas where confidential data is located and restricting the number of personnel authorized to access it can enhance the security of the system.

Confidentiality is also an important element of security engineering processes. Engineering teams must develop mechanisms that protect from unauthorized disclosure of information while still allowing authorized users to access it. Such mechanisms may involve the implementation of encryption, authentication, and access control mechanisms, as well as the use of secure coding practices to ensure that confidential information is not inadvertently disclosed through coding errors. Additionally, engineering teams must ensure that the system maintains an adequate level of confidentiality throughout its entire life cycle, as any

Warning:

If we fail to guarantee confidentiality requirements, the following could happen to the system:

- A third-party could access sensitive data stored in the system, potentially leading to unauthorized disclosures, identity theft, financial losses, and legal issues.
- Malicious actors may use the data to launch targeted attacks on the system, with the intent of disrupting business operations and gaining access to confidential information for their own gain.
- The system may become vulnerable to exploits that can be used to gain access to private data, or to interfere with its operations.
- Systems can become prone to denial of service attacks, where legitimate requests are blocked and malicious requests are allowed in order to slow down service or gain access to systems.
- Sensitive information may be exposed to unauthorized personnel, leading to the potential loss of competitive advantage in the marketplace.

Integrity

Integrity Requirement in Security Engineering

Integrity is one of the key security requirements that must be addressed in security engineering. This requirement is used to ensure the accuracy and completeness of data and systems.

The integrity requirement helps to protect against malicious attacks, such as data tampering, data manipulation, or unauthorized access. It also helps to ensure that data is kept secure and stored in its original form.

Other aspects of integrity in security engineering include:

Data authentication: Data authentication is a process of verifying the accuracy of data by validating digital signatures, checksums, encryption, and other techniques.

Access control: Access control measures help to ensure that only authorized users have access to data and systems.

Backup and recovery: Backup and recovery processes help to maintain data integrity in case of system failure or malicious attacks.

Logging and auditing: Logging and auditing are processes to track user activity and ensure data integrity.

Warning:

If we fail to guarantee integrity requirements, the system may be compromised in a variety of ways. Some possible outcomes include:

- Data can be corrupted or modified without the knowledge of the user
- Hackers can break into the system and steal sensitive information
- Malicious software can be installed in the system
- Unauthorized users may be granted access to the system
- System performance can suffer due to malicious attacks or malicious code
- Security of the system can be compromised

Availability

Availability Requirement in Security Engineering

Availability requirement in security engineering refers to the need for secure systems to remain operational and available to users when required. Achieving availability without sacrificing security is often a challenge, as attackers may attempt to disrupt system availability in order to deny service or gain access to sensitive information.

Security engineering must therefore consider and account for the availability of system components, including network connections, storage systems, and web applications. Examples of availability requirements include:

- Ensuring that system services remain available despite distributed denial of service (DDoS) or other types of attacks.
- Preventing unauthorized users from accessing systems by restricting access privileges.
- Defending against malicious code, such as viruses, worms, and Trojans.
- Developing backup strategies and business continuity plans to ensure that systems maintain acceptable levels of service.
- Measuring service availability in order to identify areas of improvement.

Availability is a key concern in security engineering

Warning:

If availability requirements are not met, then the system may suffer from:

- Decreased performance or slow response times
- Outages or downtime
- Higher than expected resource usage
- Lowered security
- Loss of data
- Increased maintenance costs

Authentication

Authentication Requirement in Security Engineering

Authentication is a security measure which requires a user to prove their identity before accessing resources or taking certain actions on a system. Authentication requirements are essential for anything related to security engineering, as they are the basis for ensuring access to the system and resources is being done by authorized personnel. Authentication requirements can typically involve one or more of the following:

Username and Password: A unique username and password combination is often the most common way to authenticate an individual. Passwords should meet the security guidelines set by the organization and must be changed regularly.

Multi-Factor Authentication: This is an additional layer of security which requires users to provide two or more pieces of evidence to prove their identity. This could include personal information such as a security code, or additional authentication methods such as biometrics or a one-time token.

PIN/Password Combo: PIN numbers may also

Warning:

If we fail to guarantee authentication requirements, the system may become vulnerable to security threats. Malicious attackers may gain unauthorized access to the system and perform malicious activities such as data theft, manipulation of data, or denial of service. This could lead to significant financial losses, reputational damage, and legal ramifications.

Authorization

Authorization Requirements in Security Engineering

Authorization requirements are security measures that ensure only authorized personnel can access a system or database. These requirements are designed to protect systems and data from malicious activity or unauthorized access. Authorization requirements include authentication mechanisms, role-based access control, and audit logging.

Authentication mechanisms are designed to ensure that users or processes are who they say they are. Authentication can be done by combining something a user knows (e.g. a password) with something they have (e.g. a token) or something they are (e.g. a biometric fingerprint scan).

Role-based access control (RBAC) enables officials to assign user roles that limit access to certain functions and data. RBAC can be used to prevent access of sensitive information to prevent data leaks or damage to the system.

Audit logging is a process of tracking and recording changes in system activities and records. Auditing logs can be used for troubleshooting

Warning:

If we fail to guarantee authorization requirements, it can lead to a number of consequences for the system:

- Unauthorized users may have access to confidential information or make changes to the system without permission.
- Data stored in the system may be manipulated or corrupted by unauthorized users.
- System performance could be significantly impacted due to malicious activity.
- System security may be compromised, resulting in a breach of sensitive information.
- Legitimate users may be denied access to the system due to incorrect permissions.

Non-repudiation

Non-repudiation is a term used in information security that refers to a legal concept describing the assurance that someone cannot deny that they performed a certain action. It is a critical security requirement for many businesses, especially in the fields of finance and e-commerce.

In security engineering, non-repudiation refers to the technical capability of preventing a source from denying having performed an action, such as sending a message or making a payment. To achieve non-repudiation, various cryptographic techniques can be used, such as digital signatures and Secure Hash Algorithm (SHA).

Non-repudiation requirement in security engineering

Non-repudiation is a critical security requirement in many organizations, as it helps ensure that the source of a transaction or message cannot be denied at a later point in time. To guarantee non-repudiation, security engineering must employ various cryptographic techniques such as digital signatures, Secure Hash Algorithm (SHA), or other methods of

Warning:

Consequences of Failing to Guarantee Non-Repudiation Requirements

If a system fails to guarantee non-repudiation requirements, it can lead to a variety of serious consequences in both the short and long term. Some of these consequences include:

- Loss of customer confidence and potential decrease in revenue due to lack of trust
- Increased risk of fraudulent activities and unauthorized transactions
- Damage to brand reputation
- Legal issues and possible fines/penalties due to non-compliance with regulations
- Inability to prove ownership or responsibility for an action
- Difficulty in resolving disputes between parties

Accountability

Accountability Requirement in Security Engineering

Security engineering is the process of designing and building secure systems. A key feature of security engineering is the requirement for accountability. This means that when something goes wrong with a system, it must be possible to determine who was responsible for the incident and take appropriate action.

Accountability has several components including:

Auditable Events: Events in the system should be logged and tracked to allow for audit and investigation.

Identification: Access controls must be in place to identify and authenticate users who interact with the system.

Authorization: Users should only be given access to resources that they have been explicitly authorized to access.

Privileges and Access Control: Access to system components must be managed and restricted to only users who have the necessary privileges and clearance.

Data Protection: Sensitive data stored within the system must be protected from

Warning:

If we fail to guarantee accountability requirements, the system will become insecure and vulnerable. This could lead to data being exposed to unauthorized persons or malicious actors. It can also lead to data breaches, where confidential and sensitive information is leaked. This could result in financial or reputational damage to the organization. Furthermore, without accountability, it can be difficult to prove who is responsible for any wrongdoing or breaches of security.

Reliability

Reliability Requirement in Security Engineering

- The system must be able to detect and record any unauthorized access attempts.
- The system must provide an adequate level of fault tolerance.
- The system must be able to inform the users of any security breaches so action can be taken.
- The system must be able to withstand natural disasters or other forms of attack.
- The system must be able to protect the confidentiality, integrity, and availability of data.
- The system must be able to detect malicious code or errors that could cause potential data loss.
- The system must be able to restore any data that is lost or corrupted in the event of an attack.
- The system must be able to notify and inform appropriate personnel of any unauthorized access attempts and malicious activity.
- The system must be able to protect itself from malicious attack and be resilient to any changes in the environment.
- The system must be designed

Warning:

If reliability requirements are not met, the system may experience decreased performance, data loss, or downtime. This could result in a loss of user confidence in the system, decreased efficiency, and potentially loss of revenue. It could also result in customers going elsewhere for services and products, leading to a decline in profits and market share.

Physical Security

Physical Security

Physical Security is the protection of people, property, and information onsite. It involves protecting physical assets from potential risks such as fire, theft, vandalism, and natural disasters.

The following should be considered when designing physical security:

- Access Control: Controlling access to the facility, equipment, resources and data with authentication mechanisms such as lock and key, bio-metric, security guards, and CCTV surveillance.
- Environmental Management: Monitoring and controlling the environmental conditions within the facility, such as temperature, humidity, fire/smoke detection, seismic activity, and water leaks.
- Emergency Response: In the event of an emergency, it is important to have comprehensive procedures in place for responding quickly and effectively.
- Equipment Protection: Protecting all hardware and critical equipment with alarms/sensors and preventing tampering.
- Systems Security: Ensuring the integrity of the digital systems and networks within the facility by implementing security measures, such as

Warning:

Potential Consequences of Failing to Guarantee Physical Security Requirements

- Theft or destruction of hardware components and systems.

- Potential exposure of confidential data or information.
- Unauthorized access to sensitive systems, networks, or data.
- Increased risk of malicious attacks.
- Increased risk of denial-of-service attacks.
- Financial losses due to equipment damage or data theft.
- Loss of customer trust, resulting in decreased or lost business.
- Legal action due to data breaches.

Forgery Resistence

Forgery Resistance is an important requirement in security engineering that aims to protect data and systems from attempts to counterfeit, clone, counterfeit, or alter the identity of an entity. It can be achieved through various means, including:

Cryptography: Cryptography is the process of transforming data into a form that only the intended recipient can read. It can prevent forgery by making it impossible for anyone to create or alter data without knowing the recipientâ€™s authentication key.

Digital Signatures: A digital signature is a way of verifying the identity of a user or verifying the integrity of a message. It uses a private/public key system to ensure that the digital signature can only be created and verified by the proper party.

Tamper-proofing: Tamper-proofing techniques such as watermarking, sealing, and inlays help prevent data from being altered or forged without authorization.

Strong Authentication: Strong authentication methods like

Warning:

If we fail to guarantee the forgery resistance requirement, the system would be vulnerable to forgery or counterfeiting of documents, which could lead to potential fraud, illegitimate access to resources, data theft, and other malicious activities. This could have serious implications for the security and integrity of the system, as well as the data it contains. Furthermore, it could open up the system to legal and financial liabilities if it is determined that the failure to guarantee forgery resistance enabled a malicious attack.

Tamper Detection

Tamper Detection

Tamper detection is a requirement in security engineering that detects and alerts for any changes made to the system. This type of security helps to ensure that confidential information is safe and not accessible to unauthorized personnel. Tamper detection technology can detect any changes made to the system such as adding or removing files, changing configurations, and more. Additionally, tamper detection can trigger other protective measures such as locking down a system or triggering alerts when a malicious attack is detected.

Warning:

If we fail to guarantee tamper detection, the security of the system can be compromised. Attackers can try to break into the system, modify data, or even inject malicious code. This can cause a variety of problems such as system crashes, data corruption, and malicious activity. Without the assurance of tamper detection, the system may be vulnerable to malicious activity, and the risk of suffering from a security breach increases.

Data Freshness

Data Freshness is a requirement in security engineering which is concerned with ensuring that data requires updating periodically and is not outdated.

In order to ensure data freshness, organizations must have a defined and enforced policy regarding when and how often the data must be updated. Some organizations may require daily or even hourly updates, while others may adopt a more relaxed approach.

Good data freshness practices also require that data must not be allowed to become stale or out-of-date, and should be regularly monitored to ensure that the data is accurate and up-to-date.

Warning:

If the system fails to guarantee data freshness, there will be a number of consequences. These include:

Unreliable data and results: Data which is not up to date can lead to unreliable insights and inaccurate business decisions.

Missed opportunities and delayed decisions: Using stale data can lead to the loss of potential opportunities, as well as a delay in making decisions.

Lack of trust: By not maintaining fresh data, the system will lose credibility with its users and may be deemed untrustworthy.

Poor customer experience: Data that is not up to date can result in a poor customer experience, leading to dissatisfaction and a loss of customers.

Confinement

Confinement requirement in security engineering

Confinement requirements in security engineering are security requirements that ensure that privileged operations and activities (both internal and external) are constrained so that they cannot be abused or manipulated for malicious purposes. These requirements are generally implemented using a combination of hardware, software, processes, policies, and other safeguards. By confining privileged operations and activities within a secure boundary and ensuring that only authorized and authenticated parties can access these operations and activities, confidential information and systems remain safe and secure.

Warning:

When confinement requirements are not met, the system can be vulnerable to security vulnerabilities and breaches. Without proper boundaries, malicious actors can have unrestricted access to the system, allowing them to tamper with data, modify settings, or take complete control over the system. This could lead to malicious activities such as unauthorized data exfiltration, espionage, and sabotage. Furthermore, if the system is not properly secured, then attackers can use this access to launch Denial-of-Service (DoS) attacks, spread malware, or install malicious software.

Interoperability

Interoperability Requirement in Security Engineering

Interoperability is an important requirement in security engineering that requires different systems and components to be able to work together. It involves the ability to exchange data, process signals, interpret codes and store information, allowing systems to interact and coordinate their actions.

Interoperability allows for the secure transfer of data between different systems, ensuring the integrity and availability of the data being exchanged. It also helps to ensure that unauthorized persons are unable to access the data and that any changes to the data can be tracked and reversed.

Interoperability also helps to ensure that the security policies of different systems are respected and enforced and that any changes in security policies are properly implemented and updated. It also helps to reduce the risks associated with implementing incompatible systems and components.

By ensuring systems are interoperable, security risks can be minimized, data can be securely transferred between different entities, and businesses can better manage and control their networks and systems.

Warning:

If we fail to guarantee interoperability requirements, the system could become unreliable. It would be difficult to connect with other systems or devices, and the system would be unable to exchange information with other systems and devices. This would lead to a lack of functionality as the system would not be able to communicate or interact with other systems and devices. Additionally, it could lead to security risks as the system would be more vulnerable to attacks and unauthorized access.

Data Origin Authentication

Data Origin Authentication

Data origin authentication is a security engineering requirement that aims to verify that data is sent securely and accurately, and that it is originating from an authenticated and trusted source. It aims to ensure that data sent from one location to another has not been modified in any way.

Data origin authentication typically involves techniques such as message authentication codes (MACs), digital signatures, and public-key infrastructure (PKI) protocols. It can also involve two-factor authentication and the use of cryptography. These techniques can be used to ensure that data is sent securely and with integrity, meaning that the data has not been tampered with or modified in transit.

Warning:

The consequences of failing to guarantee data origin authentication

- Untrusted data: Data integrity and authenticity could be compromised as untrusted sources may be allowed into the system, leading to data leakage, manipulation or other malicious activities.
- Reduced trust: Without authentication, it will be difficult to establish trust in any data or systems.
- Security breaches: It is much more likely that malicious actors could infiltrate the system and gain access to confidential information without authentication.
- Loss of data: Without authentication, there would be no way to confirm the accuracy or veracity of the data, leaving the system vulnerable to data loss.
- Increased risk: Without authentication, organizations may be more susceptible to cyber-attacks as malicious actors could easily access confidential data.