to having read and write permissions as part of the general user role. Only members of the administrator role will have all rights as a general user in addition to having permissions to execute operations."

Accountability Requirements

Accountability requirements are those that assist in building a historical record of user actions. Audit trails can help detect when an unauthorized user makes a change or an authorized user makes an unauthorized change, both of which are cases of integrity violations. Auditing requirements not only help with forensic investigations as a detective control but can also be used for troubleshooting errors and exceptions, if the actions of the software are tracked appropriately.

Auditing requirements at the bare minimum must include the following elements

- the identity of the subject (user or process) performing an action (who)
- the action (what)
- the object on which the action was performed (where)
- the timestamp of the action (when)

What is to be logged (audit trail) and what is not is a decision that is to be made in discussions with the business managers. As a best practice for security, all critical business transactions and administrative functions need to be identified and audited. Some examples of critical business transactions include the changing of the price of a product, discounts by sales agents, or changing customer banking information. The business owner should be asked for audit trail information to be incorporated into the software requirements specification. Some examples of administrative functionality include authentication attempts such as logon and logoff actions, adding a user to an administrator role, and changing software configuration.

Some good examples of accountability requirements that should be part of the software requirements are:

- "All failed logon attempts will be logged along with the timestamp and the Internet Protocol address where the request originated."
- "A before and an after snapshot of the pricing data that changed when a user updates the pricing of a product must be tracked with the following auditable fields – identity, action, object and timestamp."

- "Audit logs should always append and never be overwritten."
- "The audit logs must be securely retained for a period of 3 years."

General Requirements

Session Management Requirements

Sessions are useful for maintaining state but also have an impact on the secure design principles of complete mediation and psychological acceptability. Upon successful authentication, a session identifier (ID) is issued to the user and that session ID is used to track user behavior and maintain the authenticated state for that user until that session is abandoned or the state changes from authenticated to not-authenticated. Without session management, the user/process would be required to re-authenticate upon each access request (complete mediation) and this can be burdensome and psychologically unacceptable to the user. Since valid sessions can be potentially hijacked where an attacker takes control over an established session, it is necessary to plan for secure session management.

In stateless protocols, such as the HyperText Transport Protocol, session state needs to be explicitly maintained and carefully protected from brute force or predictable session ID attacks. In the secure software implementation chapter, we will be covering attacks on session management in more detail.

Session management requirements are those that ensure that once a session is established, it remains in a state that it will not compromise the security of the software. In other words, the established session is not susceptible to any threats to the security policy as it applies to confidentiality, integrity and availability. Session management requirements assure that sessions are not vulnerable to brute force attacks, predictability or Man-in-the-middle hijacking attempts.

Some good examples of session management secure software requirements that should be part of the requirements specifications are:

- "Each user activity will need to be uniquely tracked."
- "The user should not be required to provide user credential once authenticated within the Internet banking application."
- "Sessions must be explicitly abandoned when the user logs off or closes the browser window."
- "Session identifiers used to identify user sessions must not be passed in clear text or be easily guessable."

Errors & Exception Management Requirements

Errors & exceptions are potential sources of information disclosure. Verbose error messages and unhandled exception reports can result in divulging internal application architecture, design and configuration information. Using laconic error messages and structured exception handling are examples of good security design features that can thwart security threats posed by improper error or exception management. Software requirements that explicitly address errors and exceptions need to be defined in the software requirements documentation to avoid disclosure threats.

Some good examples of error & exception management secure software requirements that should be part of the requirements specifications are:

- "All exceptions are to be explicitly handled using try, catch and finally blocks."
- "Error messages that are displayed to the end user will reveal only the needed information without disclosing any internal system error details."
- "Security exception details are to be audited and monitored periodically."

Configuration Parameters Management Requirements

Software configuration parameters and code which makeup the software needs protection against hackers. These parameters and code usually need to be initialized before the software can run. Identifying and capturing configuration settings is vital to ensure that an appropriate level of protection is considered when the software is designed, developed and more importantly when it is deployed.

Some good examples of configuration parameters management secure software requirements that should be part of the requirements specifications are:

- "The web application configuration file must encrypt sensitive database connections settings and other sensitive application settings."
- "Passwords must not be hard-coded in line code."
- "Initialization and disposal of global variables need to be carefully and explicitly monitored."
- "Application and/or Session OnStart and OnEnd events must include protection of configuration information as a safeguard against disclosure threats"