Improper protection (initialization and enforcement)

Improper validation
- Not checking critical conditions and parameters, leading to a process addressing memory not in its memory space by referencing through an out-of-bounds pointer value, allowing type clashes, overflows.

Improper synchronization
- Not using unfair scheduling algorithms that block certain processes or users from running, using the wrong function or wrong arguments.

Improper change
- The “time-of-check to time-of-use” flaw; changing a parameter unexpectedly.

Improper deallocation or deletion
- Leaving old data in memory deallocated by one process and reallocated to another process, enabling the second process to access the information used by the first, failing to end a session properly.

Improper indivisibility
- Interrupting atomic operations (e.g., locking) due to cache inconsistency.

Improper sequencing
- Allowing actions in an incorrect order (e.g., reading during writing).

Improper choice of initial protection domain
- Incorrect initial assignment of security or integrity level at system initialization or generation; a security critical function manipulating critical data directly accessible to the user.

Improper isolation of implementation detail
- Allowing users to bypass operating system controls and write to absolute input/output addresses; direct manipulation of a hidden data structure such as a directory file being written as if it were a regular file; drawing inferences from paging activity.

Improper change
- Improper naming
- Improper deallocation or deletion
- Improper indivisibility
- Improper sequencing
- Improper choice of operand or operation
- Improper validation
- Improper protection (initialization and enforcement)