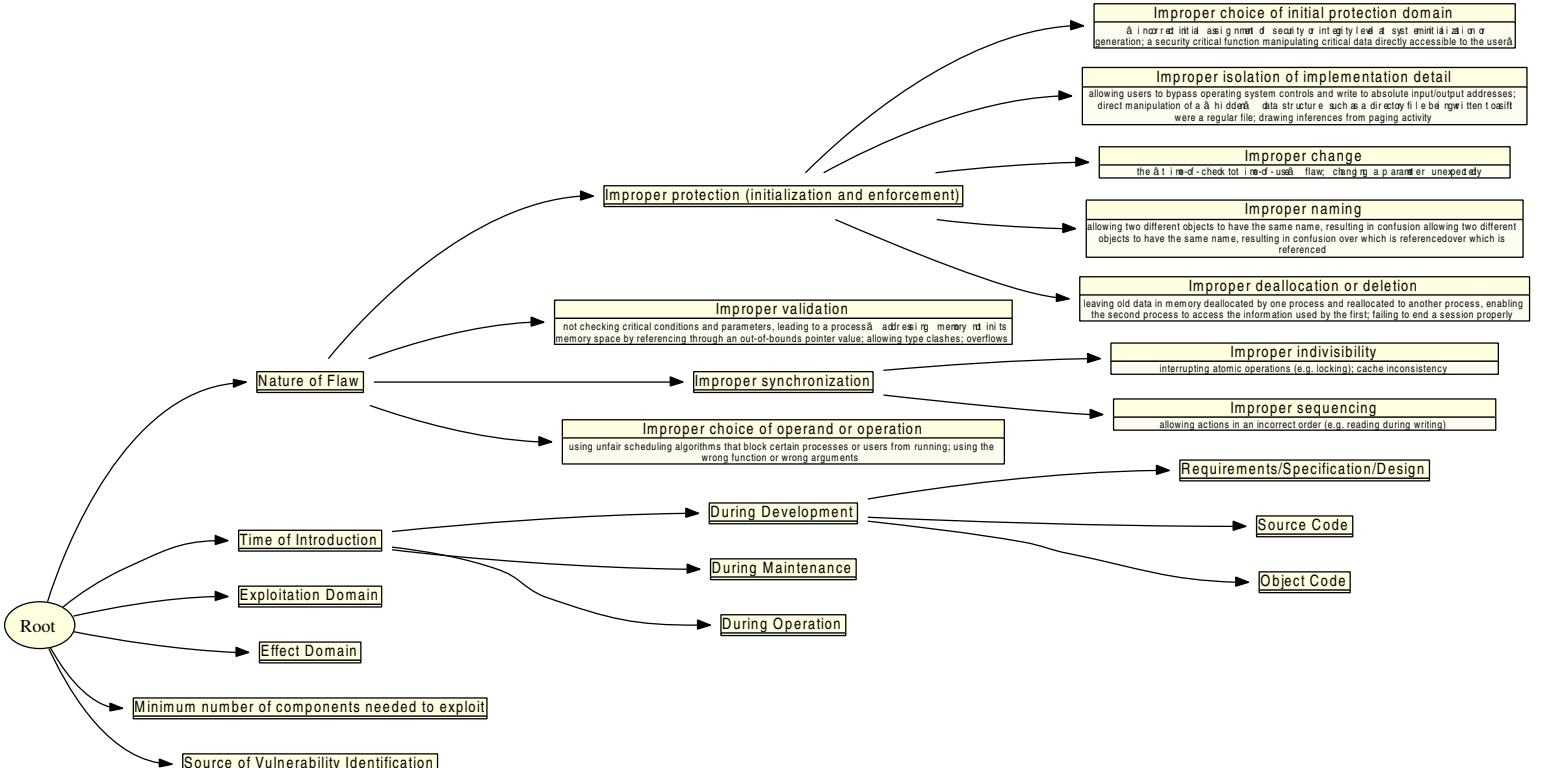
Root
Nature of Flaw
Improper protection (initialization and enforcement)
Improper choice of initial protection domain
â incorrect initial assignment of security or integrity levels at system initialization or generation; a security critical function manipulating critical data directly accessible to the userâ
Improper isolation of implementation detail
allowing users to bypass operating system controls and write to absolute input/output addresses; direct manipulation of a â hidden â data structure such as a directory file that normally treated as if were a regular file; drawing inferences from paging activity
Improper change
the â time-of-check to time-of-use â flaw; changing a parameter unexpectedly
Improper naming
allowing two different objects to have the same name, resulting in confusion allowing two different objects to have the same name, resulting in confusion over which is referencedover which is referenced
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 validation
not checking critical conditions and parameters, leading to a processâ addressing memory not in its memory space by referencing through an out-of-bounds pointer value; allowing type clashes; overflows
Improper synchronization
Improper indivisibility
interrupting atomic operations (e.g. locking); cache inconsistency
Improper sequencing
allowing actions in an incorrect order (e.g. reading during writing)
Improper choice of operand or operation
using unfair scheduling algorithms that block certain processes or users from running; using the wrong function or wrong arguments
Time of Introduction
During Development
Requirements/Specification/Design
Source Code
Object Code
During Maintenance
During Operation
Exploitation Domain
Effect Domain
Minimum number of components needed to exploit
Source of Vulnerability Identification