Memory Bugs Classes in Bugs Framework

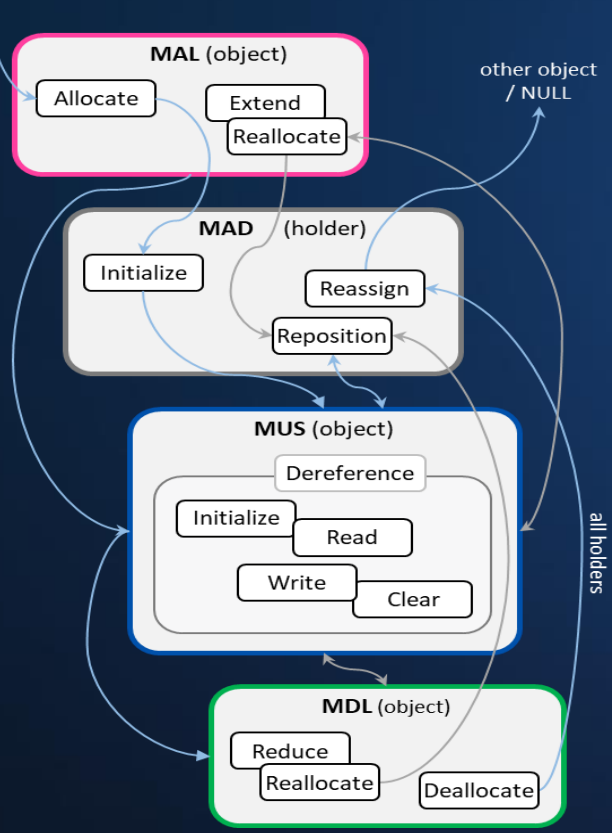
National Institute of Standards and Technology
U.S. Department of Commerce

Irena Bojanova, Carlos Galhardo

The Bugs Framework (BF) allows precise descriptions of software bugs and vulnerabilities.

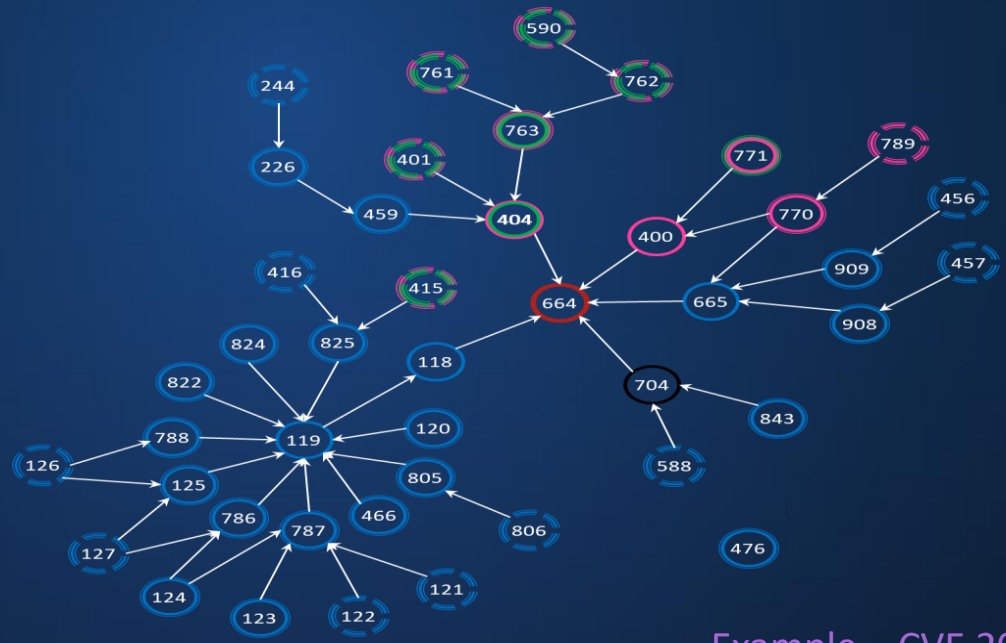
Model

Operations where bugs happen.



Mapping

BF covers all memory related CWEs and more.



Memory Deallocation Bugs Class

Causes

Improper Operation
- Missing
- Mismatched
- Erroneous

Improper Holder
- Wild Pointer
- Dangling Pointer
- Specific Address
- Wrong Position
- Forbidden Address

Improper object
- Wrong Size Used

Attributes

Operation
- Deallocate
- Reduce
- Reallocate

Source Code
- Codebase
- Third Party
- Standard Library
- Processor

Execution Space
- Userland
- Kernel
- Bare-Metal

Location
- Stack
- Heap
- (…)

 # Holders
- None
- Single
- Multiple

Consequences

Improper Holder for Next Operation
- NULL Pointer

Improper object for Next Operation
- Not Enough Allocated

Memory Error
- Memory Leak
- Double Free
- object Corruption
- Illegal Access

Example – CVE 2018:20911

In the smallvec crate for Rust, the Iterator implementation mishandles destructors, leading to a double free.

Cause

Improper Holder
- Dangling Pointer (in `SmallVec`)

MDL Attributes

Operation:
- Deallocate

Source Code:
- Standard Library (`lib.rs`)

Execution Space:
- Userland

Location:
- Heap

 # Holders:
- Multiple

Consequence

Memory Error
- Double Free

- Initial state: V = [a, b, c]
- Intermediate state: V = [a, b, b, c]
- If the iterator panics, V deallocates.

after "fix"

Cause

Improper object
- Wrong Size Used (for `SmallVec`)

MDL Attributes

Operation:
- Deallocate

Source Code:
- Standard Library (`lib.rs`)

Execution Space:
- Userland

Location:
- Heap

 # Holders:
- Multiple

Consequence

Memory Error
- Memory Leak

Corresponding BF classes:

- Memory Allocation (MAL)
- Memory Addressing (MAD)
- Memory Use (MUS)
- Memory Deallocation (MDL).