Memory Bugs Classes in NIST Bugs Framework (BF)

Handouts

Irena Bojanova, NIST

Carlos Galhardo, INMETRO

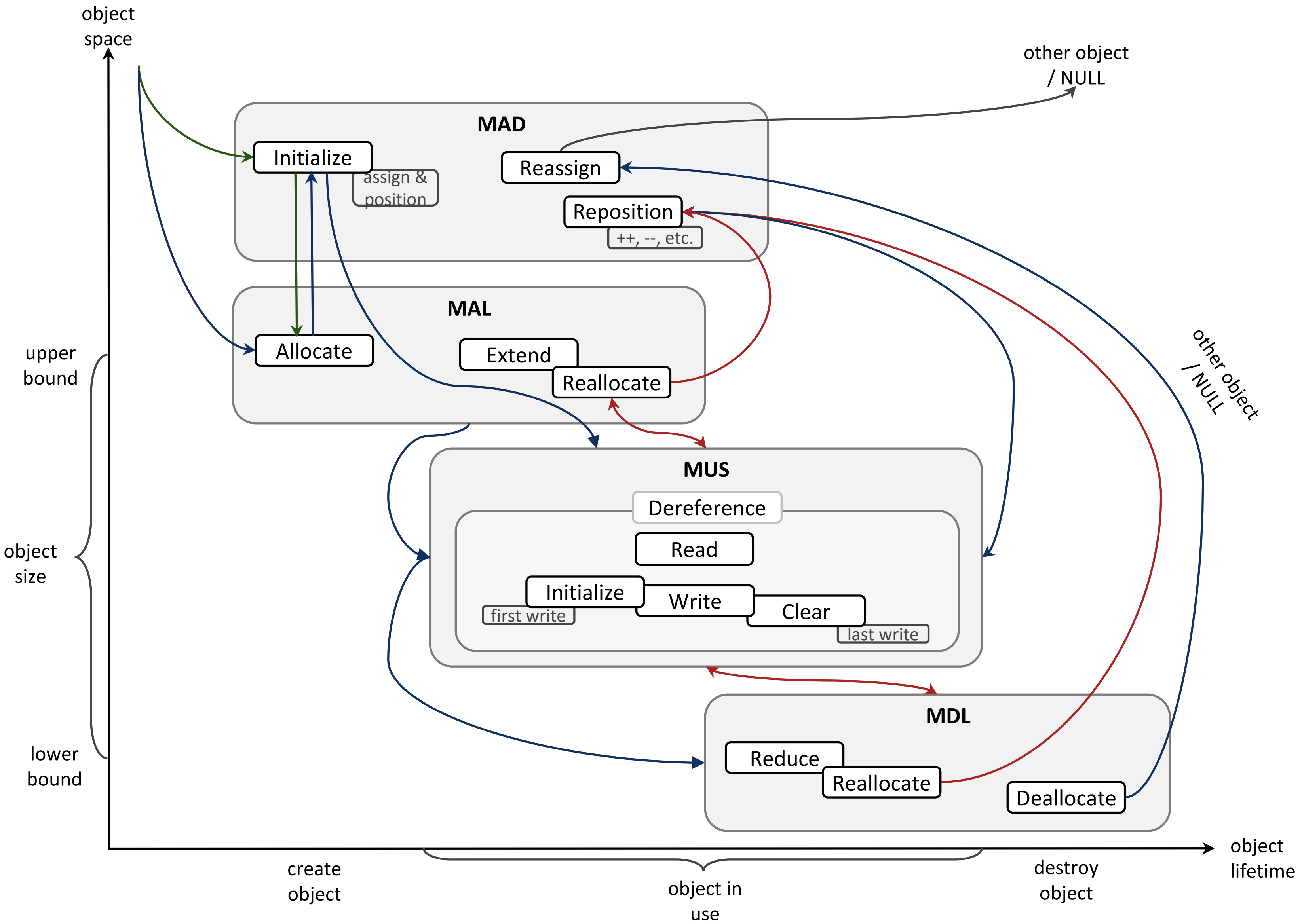
National Institute of Standards and Technology
U.S. Department of Commerce

SAMATE

INMETRO

HCSS 2020
09/15/2020

The BF Memory Bugs Model



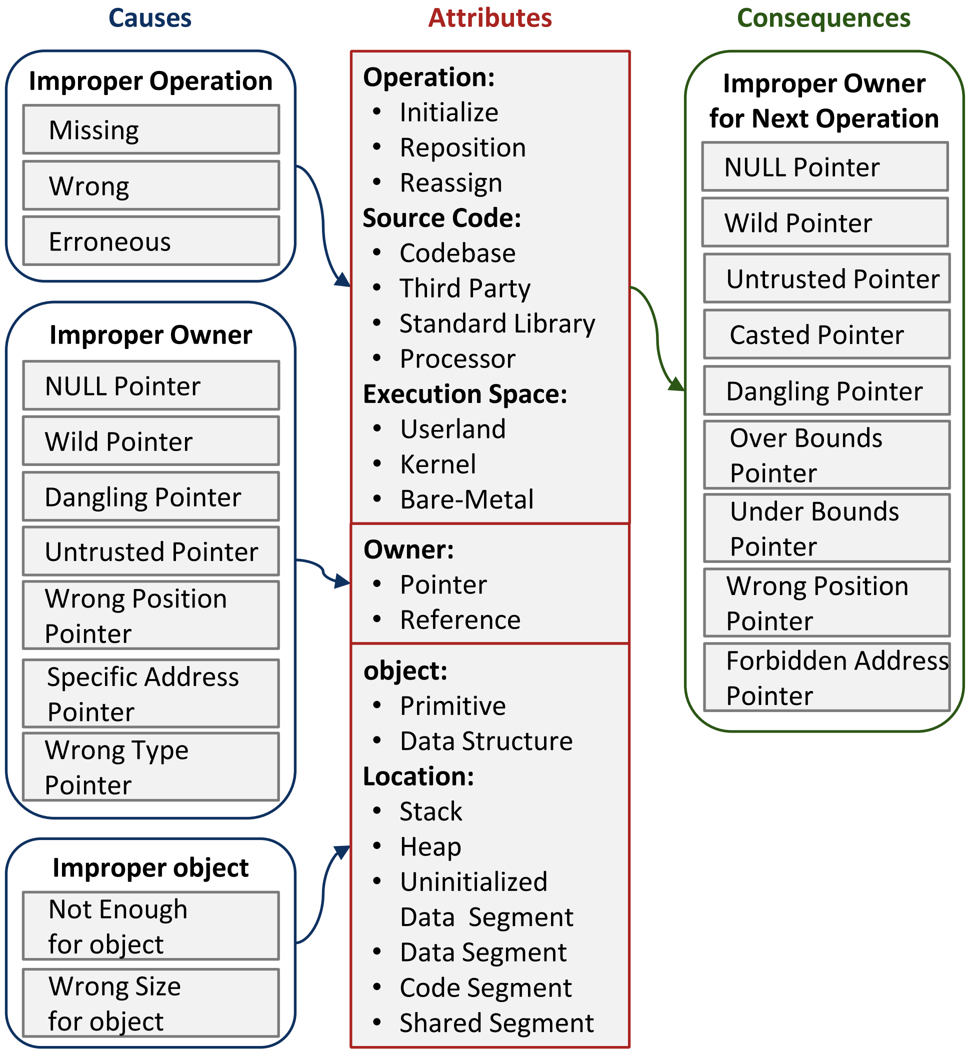
MAD – Memory Addressing Bugs

MAL – Memory Allocation Bugs

MUS – Memory Use Bugs

MDL – Memory Deallocation Bugs

MAD – Memory Addressing Bugs



Causes

Improper Operation
- Missing
- Wrong
- Erroneous

Improper Owner
- NULL Pointer
- Wild Pointer
- Dangling Pointer
- Untrusted Pointer
- Wrong Position Pointer
- Specific Address Pointer
- Wrong Type Pointer

Improper object
- Not Enough for object
- Wrong Size for object

Attributes

Operation:
- Initialize
- Reposition
- Reassign

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

Execution Space:
- Userland
- Kernel
- Bare-Metal

Owner:
- Pointer
- Reference

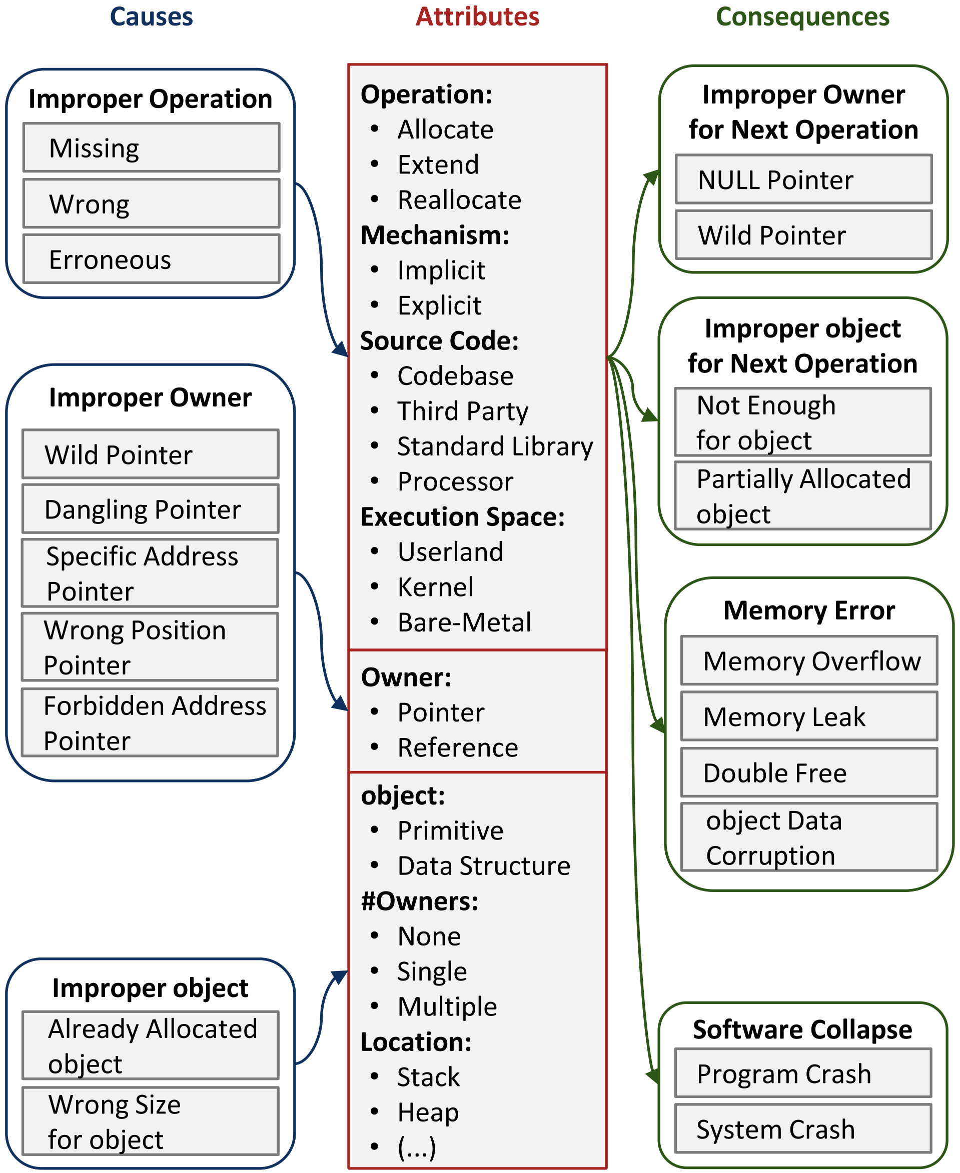
object:
- Primitive
- Data Structure

Location:
- Stack
- Heap
- Uninitialized Data Segment
- Data Segment
- Code Segment
- Shared Segment

Consequences

Improper Owner for Next Operation
- NULL Pointer
- Wild Pointer
- Untrusted Pointer
- Casted Pointer
- Dangling Pointer
- Over Bounds Pointer
- Under Bounds Pointer
- Wrong Position Pointer
- Forbidden Address Pointer

MAL – Memory Allocation Bugs



Causes

Improper Operation
- Missing
- Wrong
- Erroneous

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

Improper object
- Already Allocated object
- Wrong Size for object

Attributes

Operation:
- Allocate
- Extend
- Reallocate

Mechanism:
- Implicit
- Explicit

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

Execution Space:
- Userland
- Kernel
- Bare-Metal

Owner:
- Pointer
- Reference

object:
- Primitive
- Data Structure

#Owners:
- None
- Single
- Multiple

Location:
- Stack
- Heap
- (...)

Consequences

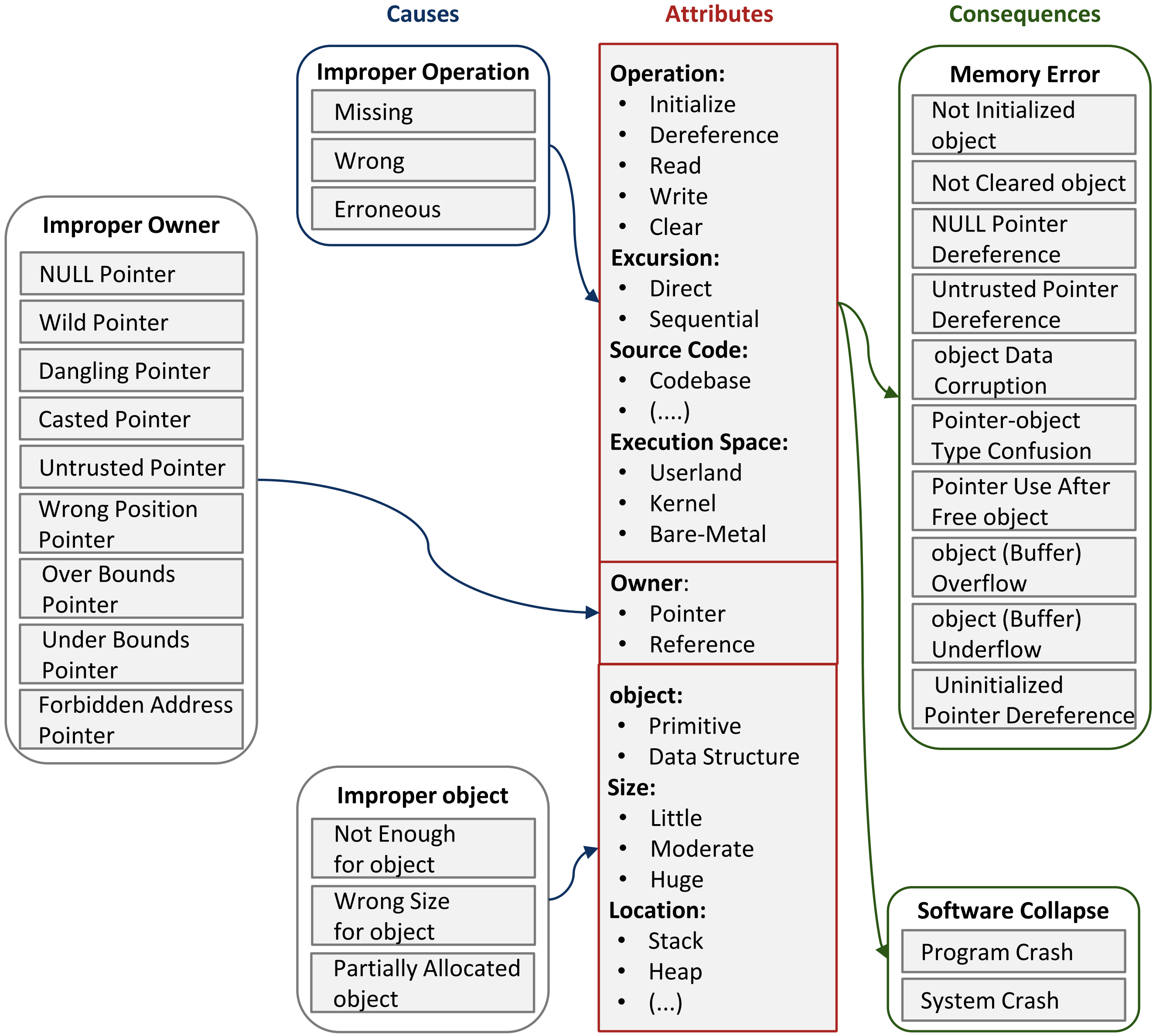
Improper Owner for Next Operation
- NULL Pointer
- Wild Pointer

Improper object for Next Operation
- Not Enough for object
- Partially Allocated object

Memory Error
- Memory Overflow
- Memory Leak
- Double Free
- object Data Corruption

Software Collapse
- Program Crash
- System Crash

MUS – Memory Use Bugs



Causes

Improper Operation
- Missing
- Wrong
- Erroneous

Improper Owner
- NULL Pointer
- Wild Pointer
- Dangling Pointer
- Casted Pointer
- Untrusted Pointer
- Wrong Position Pointer
- Over Bounds Pointer
- Under Bounds Pointer
- Forbidden Address Pointer

Improper object
- Not Enough for object
- Wrong Size for object
- Partially Allocated object

Attributes

Operation:
- Initialize
- Dereference
- Read
- Write
- Clear

Excursion:
- Direct
- Sequential

Source Code:
- Codebase
- (....)

Execution Space:
- Userland
- Kernel
- Bare-Metal

Owner:
- Pointer
- Reference

object:
- Primitive
- Data Structure

Size:
- Little
- Moderate
- Huge

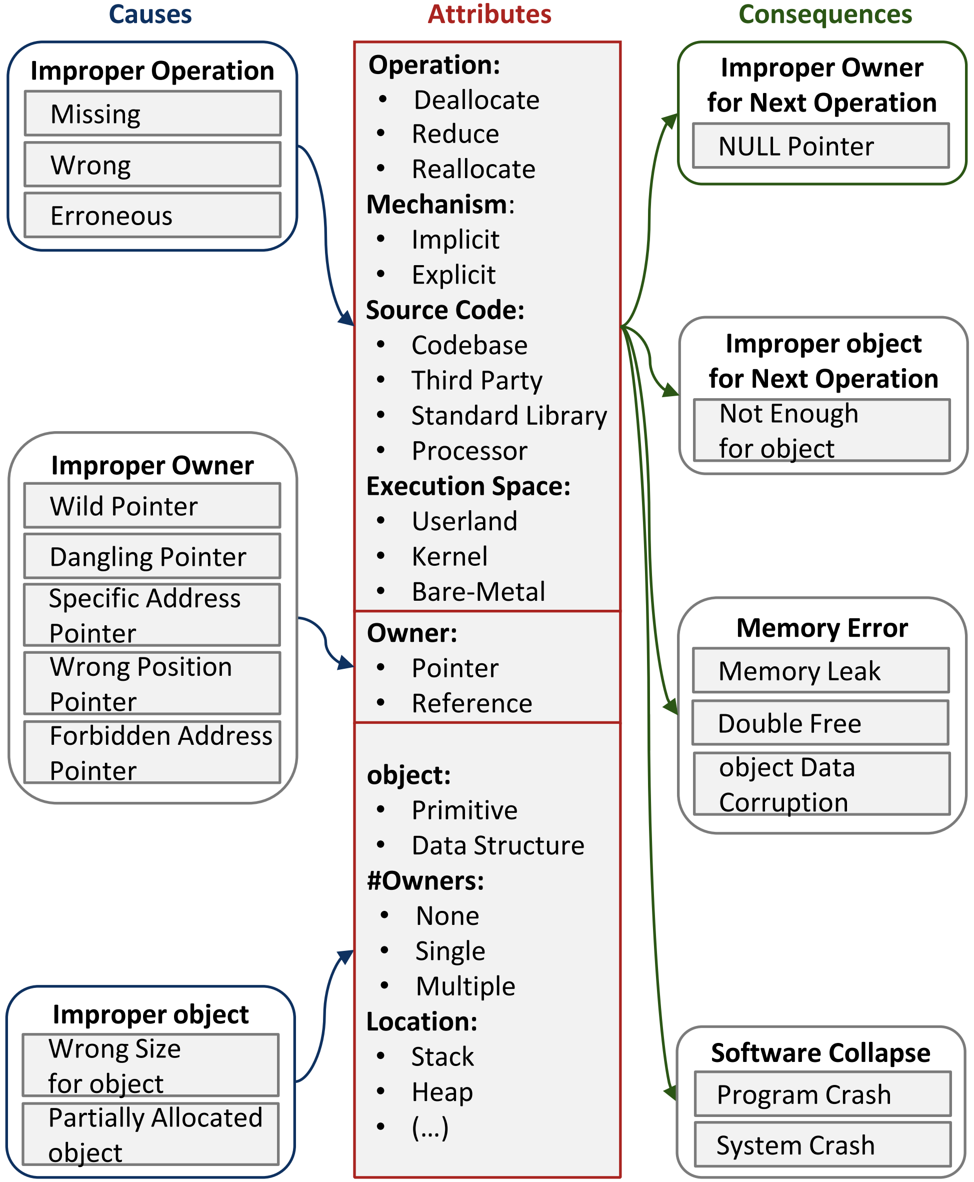
Location:
- Stack
- Heap
- (...)

Consequences

Memory Error
- Not Initialized object
- Not Cleared object
- NULL Pointer Dereference
- Untrusted Pointer Dereference
- object Data Corruption
- Pointer-object Type Confusion
- Pointer Use After Free object
- object (Buffer) Overflow
- object (Buffer) Underflow
- Uninitialized Pointer Dereference

Software Collapse
- Program Crash
- System Crash

MDL – Memory Deallocation Bugs



Causes

Improper Operation
- Missing
- Wrong
- Erroneous

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

Improper object
- Wrong Size for object
- Partially Allocated object

Attributes

Operation:
- Deallocate
- Reduce
- Reallocate

Mechanism:
- Implicit
- Explicit

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

Execution Space:
- Userland
- Kernel
- Bare-Metal

Owner:
- Pointer
- Reference

object:
- Primitive
- Data Structure

#Owners:
- None
- Single
- Multiple

Location:
- Stack
- Heap
- (…)

Consequences

Improper Owner for Next Operation
- NULL Pointer

Improper object for Next Operation
- Not Enough for object

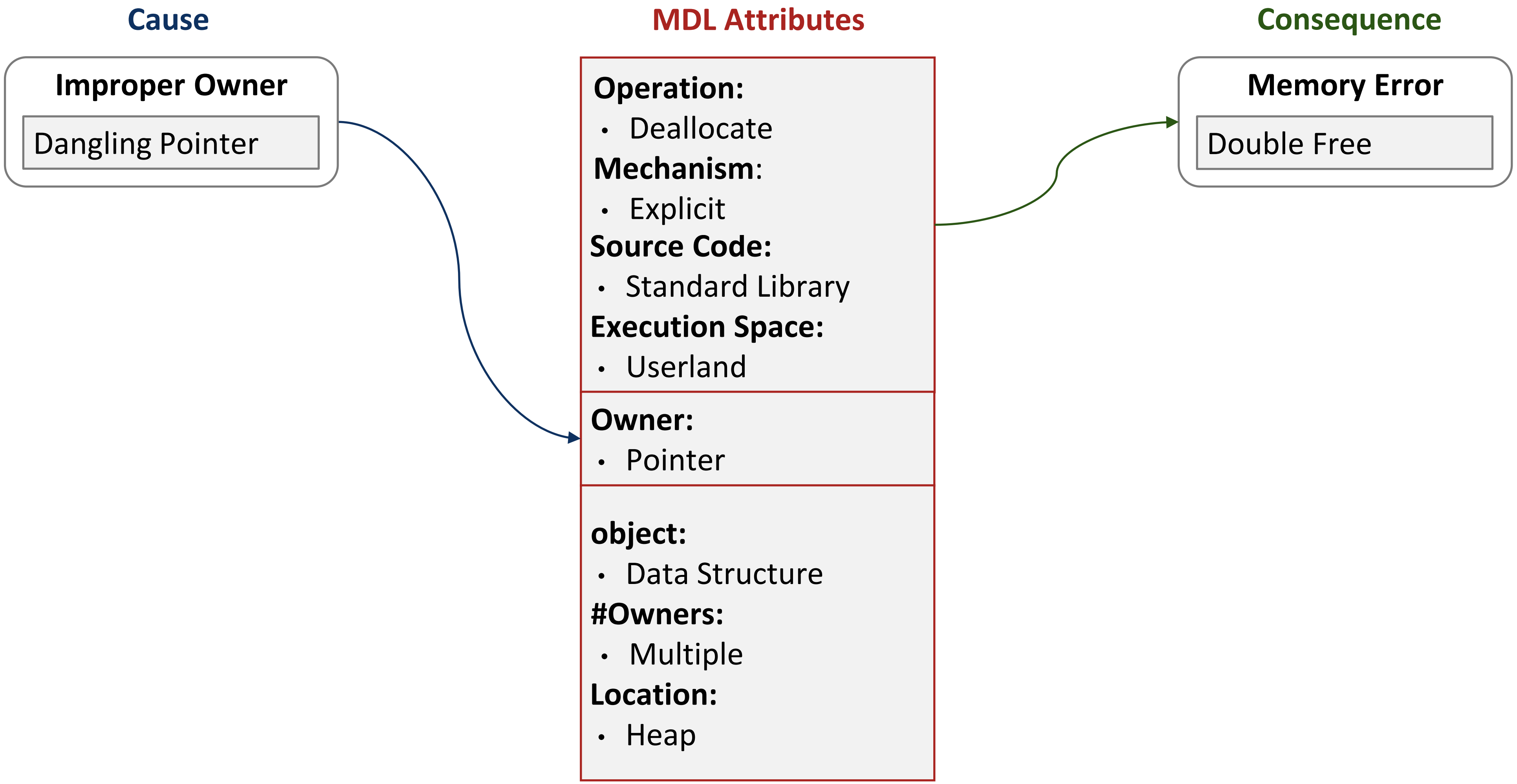
Memory Error
- Memory Leak
- Double Free
- object Data Corruption

Software Collapse
- Program Crash
- System Crash

Example 1: CVE-2018-20991

- CVE description: An issue was discovered in the smallvec crate before 0.6.3 for Rust. The Iterator implementation mishandles destructors, leading to a double free.

BF MDL Description of CVE-2018-20991



Cause

Improper Owner

Dangling Pointer

MDL Attributes

Operation:
- Deallocate

Mechanism:
- Explicit

Source Code:
- Standard Library

Execution Space:
- Userland

Owner:
- Pointer

object:
- Data Structure

#Owners:
- Multiple

Location:
- Heap

Consequence

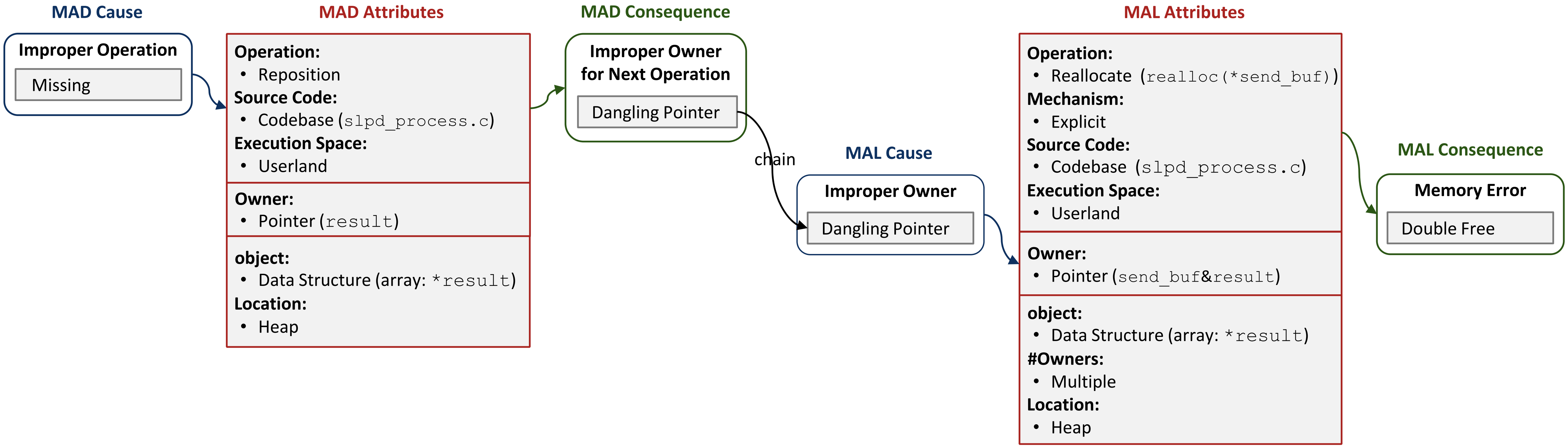
Memory Error

Double Free

Example 2: CVE-2017-17833

- CVE description: OpenSLP releases in the 1.0.2 and 1.1.0 code streams have a heap-related memory corruption issue which may manifest itself as a denial-of-service or a remote code-execution vulnerability.

CVE-2017-17833



MAD Cause

Improper Operation

Missing

MAD Attributes

Operation:
- Reposition

Source Code:
- Codebase (`slpd_process.c`)

Execution Space:
- Userland

Owner:
- Pointer (`result`)

object:
- Data Structure (array: `*result`)

Location:
- Heap

MAD Consequence

Improper Owner for Next Operation

Dangling Pointer

chain

MAL Cause

Improper Owner

Dangling Pointer

MAL Attributes

Operation:
- Reallocate (`realloc(*send_buf)`)

Mechanism:
- Explicit

Source Code:
- Codebase (`slpd_process.c`)

Execution Space:
- Userland

Owner:
- Pointer (`send_buf&result`)

object:
- Data Structure (array: `*result`)

#Owners:
- Multiple

Location:
- Heap

MAL Consequence

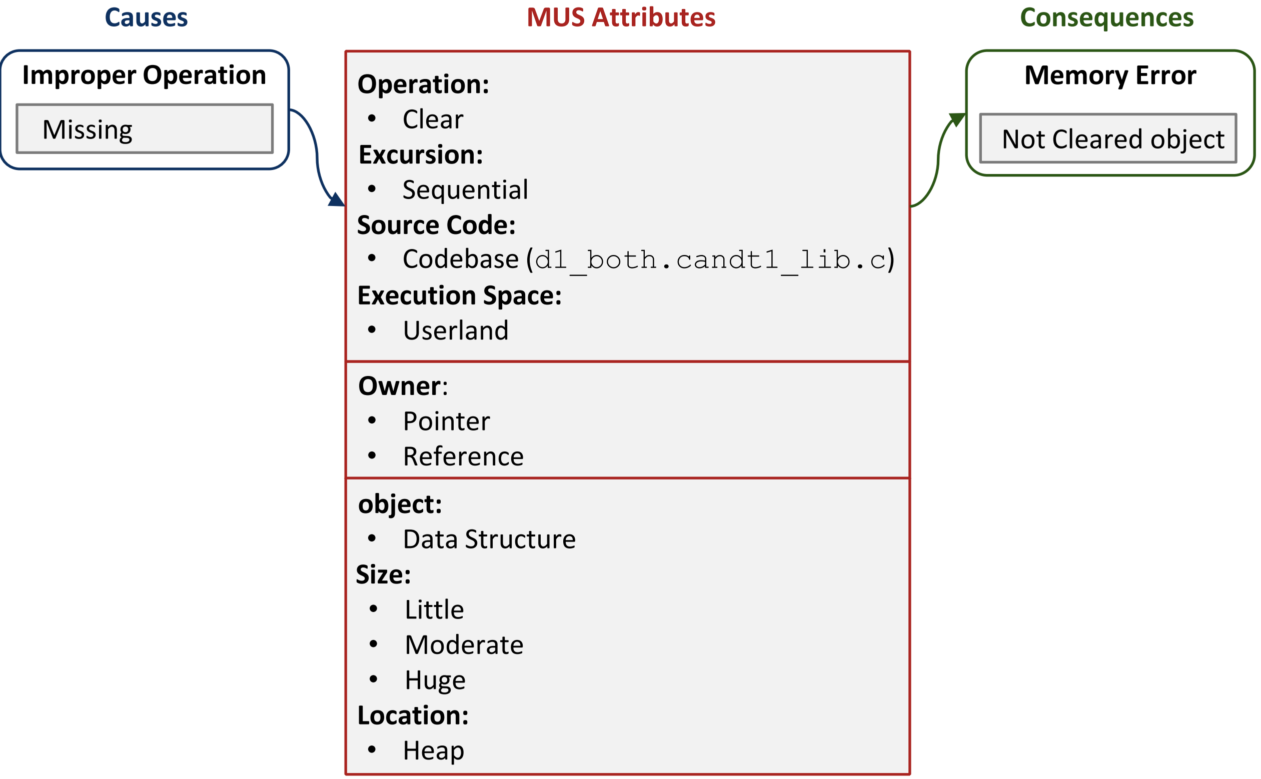
Memory Error

Double Free

Example 3: CVE-2014-0160 – Heartbleed

- CVE description: The (1) TLS and (2) DTLS implementations in OpenSSL 1.0.1 before 1.0.1g do not properly handle Heartbeat Extension packets, which allows remote attackers to obtain sensitive information from process memory via crafted packets that trigger a buffer over-read, as demonstrated by reading private keys, related to d1_both.c and t1_lib.c, aka the Heartbleed bug.

CVE-2014-0160 – Heartbleed



Causes

Improper Operation

Missing

MUS Attributes

Operation:
- Clear

Excursion:
- Sequential

Source Code:
- Codebase (`d1_both.candt1_lib.c`)

Execution Space:
- Userland

Owner:
- Pointer
- Reference

object:
- Data Structure

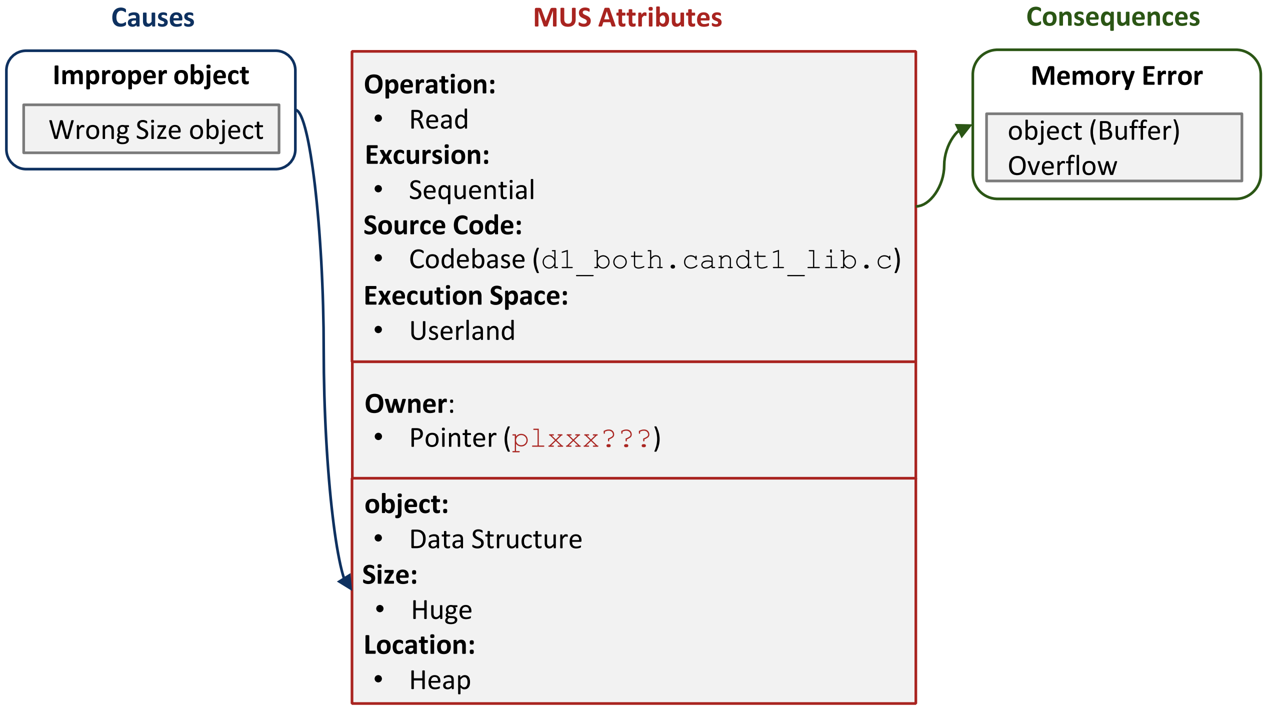
Size:
- Little
- Moderate
- Huge

Location:
- Heap

Consequences

Memory Error

Not Cleared object



Causes

Improper object

Wrong Size object

MUS Attributes

Operation:
- Read

Excursion:
- Sequential

Source Code:
- Codebase (`d1_both.candt1_lib.c`)

Execution Space:
- Userland

Owner:
- Pointer (`plxxx???`)

object:
- Data Structure

Size:
- Huge

Location:
- Heap

Consequences

Memory Error

object (Buffer) Overflow

Contact Us

irena.bojanova@nist.gov

cegalhardo@inmetro.gov.br

BF Web Site: https://samate.nist.gov/BF/