Towards a "Periodic Table" of Bugs

Paul E. Black Irena Bojanova Yaacov Yesha Yan Wu

{paul.black, irena.bojanova, yaacov.yesha}@nist.gov yanwu@bgsu.edu

Problem: Existing classifications must be improved:

- Common Weakness Enumerations (CWEs) are:
  - ✓coarse-grained and not orthogonal
- Software Fault Patterns don't include:
  - ✓attacks, upstream influences or consequences
- Semantic Templates are:
  - ✓only general interactions.

Solution: A formal orthogonal "periodic table" of software weaknesses (bugs) that :

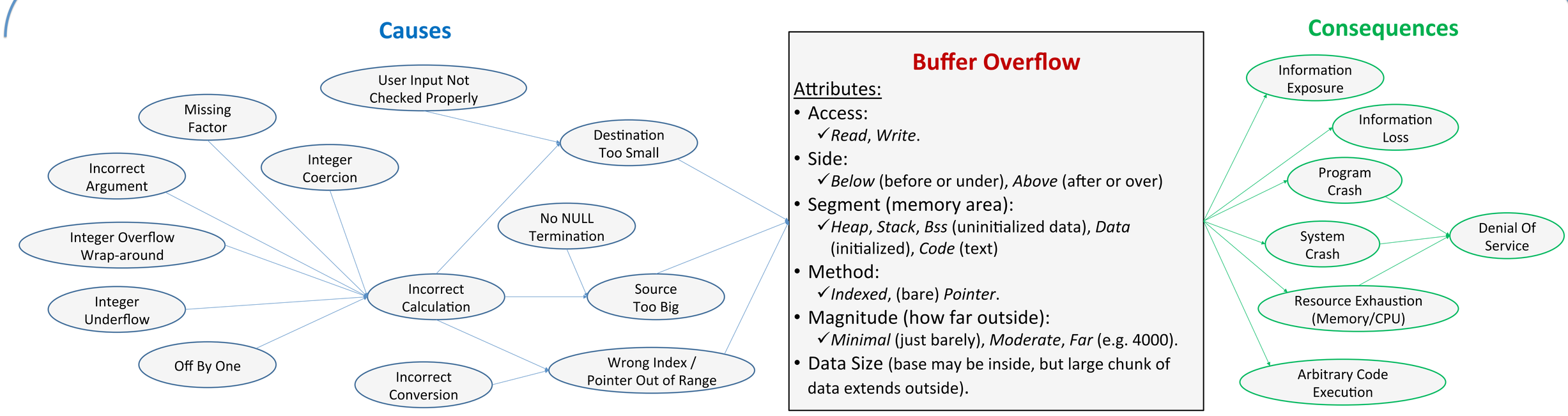
- Allows one to more closely describe:
  - ✓ what weaknesses class a tool warning covers
  - ✓ the nature of a vulnerability (e.g. Heartbleed, Shellshock, Ghost, etc.)
- Eliminates the need for an exhaustive Cartesian product of weakness classes in the CWE.

OS Command Injection (e.g. CWE-78): For a common trusted input and two untrusted inputs, the sub-sequences of code symbols in the output program differ in a way that is not included in a description of a given syntax of allowed different sequences.

Buffer Overflow (e.g. CWE-119): The software can access through a buffer a memory location that is not allocated to that buffer.

Allowing Excessive Authentication Attempts (e.g. CWE-307): The software does not limit the number of failed authentication attempts or may allow more than a specified number of failed authentication attempts within a specified time period.

*Current CWE Definitions are at the bottom



Example: Ghost (CVE-2015-0235) — glibc gethostbyname buffer overflow is

- caused by a *Destination Too Small*
- because of an *Incorrect Calculation* specifically *Missing Factor*
- where there was a *Write* that was *After* the end by a *Moderate* number of bytes
- of a buffer in the *Heap*
- which may be exploited for *Arbitrary Code Execution*.

Example: Chrome WebCore (CVE-2010-1773) — toAlphabetic render buffer overflow is

- caused by a *Wrong Index*
- because of an *Incorrect Calculation* specifically *Off by One*
- where there was a *Read* that was *Below* the start by a *Minimal* amount
- of a buffer in the *Heap*
- which leads to use of *User Input Not Checked Properly*
- which may be exploited for *Information Exposure*, *Arbitrary Code Execution*, or *Program Crash* leading to *Denial of Service.*

Example: cppCheck Warning Classes

Warning\Attribute	Access	Side	Indexed	Size	Magnitude
array Index Out Of Bounds	-	-	Yes	-	-
buffer Access Out Of Bounds	-	-	-	-	-
out Of Bounds	-	-	-	-	-
negative Index	-	Below	Yes	-	-
insecure Cmd Line Args	-	-	-	-	-
write Outside Buffer Size	Write	-	-	-	-
invalid Scanf	Write	Above	-	Variable	Moderately outside

By the way, Heartbleed is *not* a buffer overflow. It involves:

- CWE-20: Improper Input Validation
- CWE-908 Use of Uninitialized Resource (memory)
- leaving sensitive information in memory (perhaps CWE-244: Improper Clearing of Heap Memory Before Release ('Heap Inspection') or CWE-226: Sensitive Information Uncleared Before Release)
- CWE-201: Information Exposure Through Sent Data.

Current CWE Definitions followed by our comments

CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection'): The software constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component.
→ "Using input", "intended command", and "correctly neutralizing" are imprecise. Our definition precisely defines "using input" and "intended command". We do not include "correctly neutralizing", because it simply means that intended OS command cannot be modified.

CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer: The software performs operations on a memory buffer, but it can read from or write to a memory location that is outside of the intended boundary of the buffer.
→ "Read from or write to a memory location" is not tied to the buffer. Our definition clarifies that access is through the same buffer to which the intended boundary pertains. Our definition also accurately, precisely, and concisely describes violation of memory safety.

CWE-307: Improper Restriction of Excessive Authentication Attempts: The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks.
→ "Multiple" and "short" are vague. Our definition recognizes that CWE-307 actually represents a set of weaknesses, each of which satisfies particular institution-specific definitions of "multiple" and "short".

National Institute of Standards and Technology