CWE User Experience Working Group Meeting Notes

August 28, 2024

Participants

- Chris Coffin
- Alec Summers
- Steve Christey Coley
- Matt DiVisconte
- Lisa Olson
- Matthew Coles
- John Keane
- Przemyslaw (Rogue) Roguski
- Christopher Sundberg
- Faheem Ahmed
- Paul Wortman
- Doug Nichols
- Kelsi Gardner
- Abhi Balakrishnan
- Tori Mills

Agenda

- CWE User Interviews
- Buffer Overflow Definition and Usage
- Open Discussion

Action Items

- Brainstorm on the discussion points made during the meeting and discuss with the community on how to move forward regarding weakness naming issues and inconsistencies.

Meeting Summary

- **Update on User Interviews**: Chris Coffin reported on the development of a User Interview Guide and the initial outreach to CWE community members for live interviews. There is potential for expanding this effort to a broader user base.
- **Buffer Overflow Terminology**: Steve Christey Coley highlighted the inconsistent usage of the term "Buffer Overflow," noting its varied definitions and implications for CWE usability. John Keane's research identified 15 related terms, underscoring the need for clarity.
- **Proposals for Terminology Consistency**: Two proposals were discussed: using "buffer overflow" to mean "write past the end of the buffer" for general usability and avoiding the term altogether in favor of precise "read/write before/after" descriptions. Each has its advantages and drawbacks.

- **Need for Broader Reorganization**: The discussion acknowledged that terminology inconsistencies extend beyond buffer overflows to other areas like integer wraparound and memory leaks. A broader CWE reorganization is necessary to address these issues comprehensively.

Meeting Notes

Update on the CWE User Interviews (Chris Coffin)

- Developed a User Interview Guide to gather feedback.
- Identified and emailed an initial set of users from the CWE community to participate in a live interview based on the User Interview Guide.
    - This may get expanded to include a broader set of users.

Buffer Overflow Definition, Usage, and Handling Terminology Differences (Steve Christey Coley)

- Inconsistent Usage of "Buffer Overflow" Term
    - "Buffer Overflow" has been used for decades, yet there isn't a single definition that's shared by everybody using it.
    - This has CWE usability implications, especially if CWE uses the term. There are many different instances in which the term is used mostly with operation and position.
    - The simplification of CWEs is becoming more and more prominent so the usage of the term is becoming more important.
    - John Keane: Did some extensive research on this topic and came up with 15 different terms related to "buffer overflow". One of the doctors he referenced worked on something similar related to bug taxonomy.
    - **Overall, people have different uses for this term which ultimately leads to inconsistencies for how people think about and use buffer overflow.**
    - In early CWE days, multiple code analysis tool vendors said that their tools do not distinguish between before/after or do not distinguish between read/write.
    - "AddressSanitizer" tool from Google flags any out-of-bounds accesses, but it reports "heap-buffer-overflow" and "stack-buffer-overflow" but doesn't report "underflows".
- CWE-122: Heap-based Buffer Overflow
    - One of the most frequently accessed CWEs.
    - "A heap-overflow condition is a buffer overflow, where the buffer that can be overwritten is allocated in the heap portion of memory."
    - Child of CWE-787: out-of-bounds write
        - "The product writes data past the end, or before the beginning, of the intended buffer."
    - CWE's own language is inconsistent. CWEs own buffer overflow model has gaps and overlap.
    - There are some cases in buffer memory operations where one can map two different CWEs and be correct based on interpretation of those CWEs.
- Discussion: What to do?

    - Some CWE reorganization will be needed regardless; this discussion is only focused on terminology.
    - Challenge: most users will not consider CWEs own definition if they are already familiar with a term.
        - Proposal 1: since most people interpret "buffer overflow" as "write past the end of the buffer" – use this term when applicable.
            - This is more usable for most people, but loses the technical accuracy for experts, exploit writers, and tool vendors.
        - Proposal 2: avoid using "buffer overflow" in names and descriptions and use less ambiguous "read/write" for "before/after" the buffer.
            - More technical accurate and precise and reduces ambiguity but doesn't use language that many people are familiar with.
        - Alternate terms/glossary could be updated when needed.
        - CWE-122, CWE-121 (stacked-based overflow), CWE-120 ("classic" buffer overflow) and others may need modifications.
- Once we have a strategy as to what to do with these terminology inconsistencies, we would apply it and fix it appropriately across the broad range of CWE.
- It's not just buffer overflows, we will most likely run into this issue as we move through CWE for simplification.
    - Other examples:
        - Integer "wrap" (wraparound) and "overflow" are technically distinct.
        - "Stack overflow" could be stack exhaustion or stack-based buffer overflow.
        - "Memory leak" could not be freeing memory after use or disclosing the contents of memory.
        - "Named callable" – formally defined as a CWE term used in standards that generalize concepts like functions, procedures, subroutines, etc.
        - "Neutralization" and other terms that CWE had to "invent".
        - IDOR/BOLA
- What is the proper way to define some of these core topics like buffer overflow and other things?