Examining Memory Safety Regressions in Open Source Software

Andy Sadler

Rose-Hulman Institute of Technology & Ulm University of Applied Sciences

Introduction

Is the rate of memory safety regressions increasing or decreasing?

Memory safety bug: a bug that happens by violating various system invariants regarding memory

Regression: a bug that comes back

Regression rate depends on type of memory safety violation

- *DieHard* paper by Berger & Zorn
- *Characterizing and predicting which bugs get reopened* by Zimmermann et. al.

Hypothesis

The rate is increasing

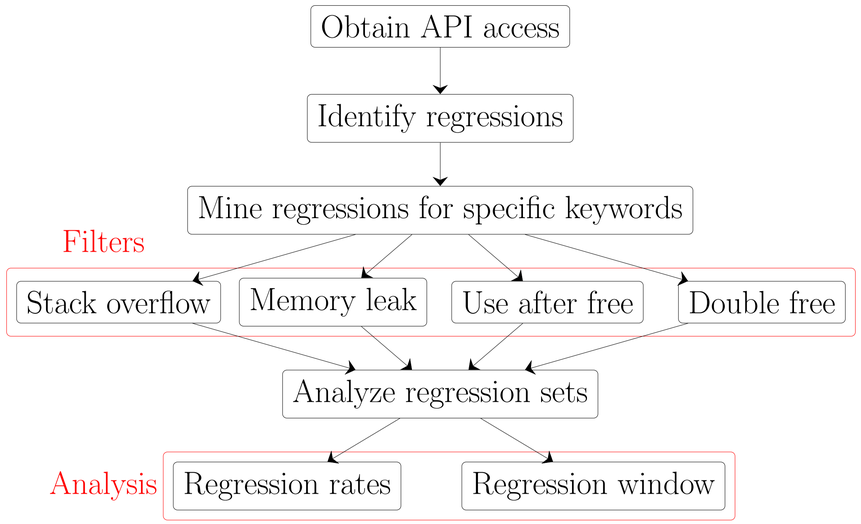
- Tools to detect these bugs have gotten better, so we're able to detect these regressions faster
- We have more memory-unsafe code written than before, so we just have more bugs to begin with

Methodology

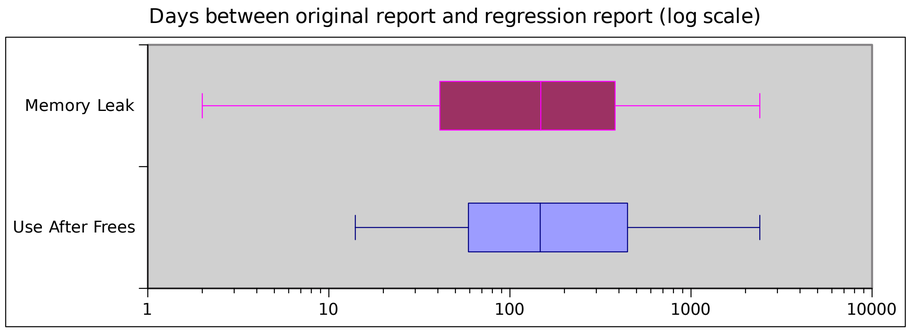
Mine bug databases for memory safety regressions

- Large project with significant bug history
- Written in memory-unsafe language, such as C or C++
- Publicly accessable bug tracker

Architecture

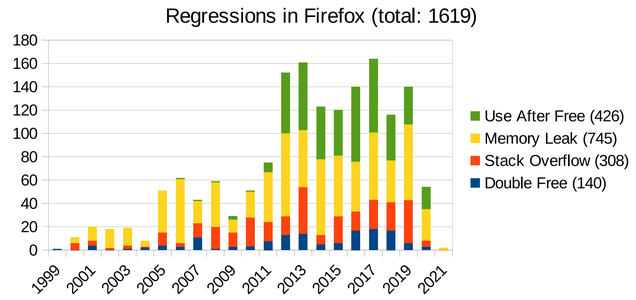


Regression window



50% occur by 150 days after, and 75% occur by 500 days after

Regression Rates



Conclusions

- Regressions have increased, but they have decreased withing the last year
- 75% probability to reappear within 500 days
- Tooling has gotten better, so we're able to detect them better
  - Valgrind (circa 2006)
  - AddressSanitizer (circa 2012)

This confirms my hypothesis.