Vulnerability Discovery

Current vulnerability discovery techniques such as black-box fuzz testing and concolic testing are so effective that they routinely find hundreds of thousands of crashers, which crash the target program. We created a new methodology for precisely and naturally defining vulnerabilities through the creation of patches. We use our methodology to debunk three commonly held beliefs in fuzzing practice.

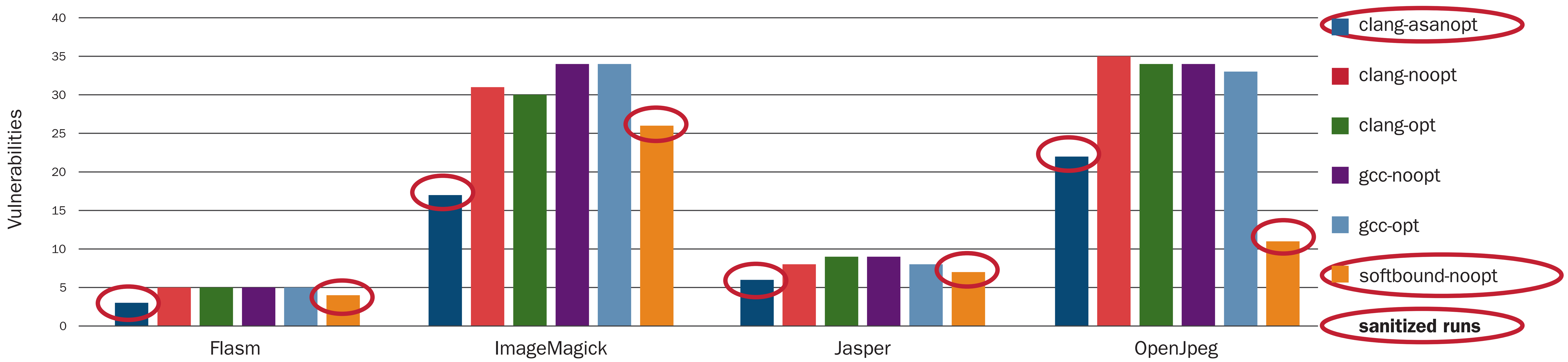
Experiment setup.
We fuzzed Flasm, ImageMagick, Jasper, and OpenJpeg for a week under various configurations, which yielded hundreds of thousands of crashes. We patched each crash using our methodology, which yielded vulnerabilities for each program. We used this data to debunk the following beliefs shown on the right:

Misbelief 1: Stack backtrace hashing always accurately counts vulnerabilities

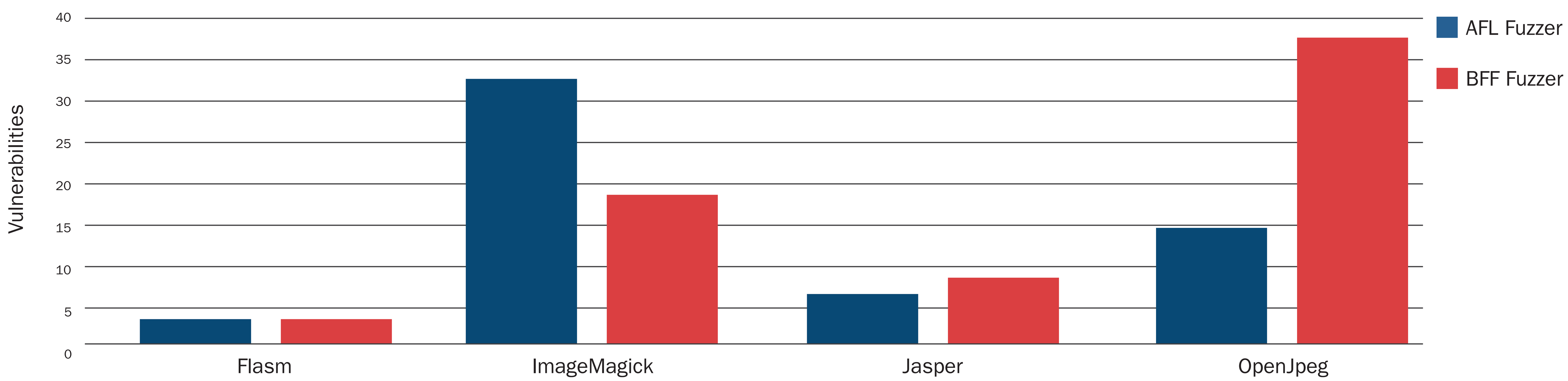
- # Vuls: Number of vulnerabilities as counted by our methodology
- UC (Undercount): Average number of vuls missed due to stack backtrace hashing
- OC (Overcount): Average number of vuls counted more than once by stack backtrace hashings

Program	# Vuls	UC	% Error	OC	% Error
Flasm	6	1.8	29%	410.9	6,848%
ImageMagic	31	1.9	6%	67.9	219%
Jasper	12	0.0	0%	226.4	1,887%
OpenJpeg	36	0.1	0%	267.5	743%

Misbelief 2: Sanitization never harms fuzzing performance



Misbelief 3: The AFL fuzzer always finds more vulnerabilities than non-guided fuzzers



Contact: David Warrren | dwarren@sei.cmu.edu