Effects of Workload Section

1 Effects of Workload Selection

As mentioned in Section "Capture Dynamic Patterns of Checks" of our submitted paper, SanRazor relies on dynamic coverage patterns to pinpoint potentially redundant checks. Therefore, in this section, we present study and discussion on the efficiency of check reduction w.r.t. the size of workload. To do so, we incrementally enlarge the workload for profiling `bzip2` record both the number of reduced sanitizer checks and the runtime overhead caused by ASan. For this study, we configure SanRazor with the *L2* scheme for check reduction. We download a `bzip2` testsuite with 38 different inputs from [2] and conduct experiments on `bzip2` (a single-file version from [1]).

As illustrated in Fig. 1, the percentage of reduced sanitizer checks will increase when more inputs are fed to the test case. However, when the number of test inputs are more than eight, the number of reduced checks reach to the saturation point at 58.0%. Similarly, the percentage of reduced runtime cost can also become stable when the number of inputs is larger than 12. Also, notice that the blue line will first increase (when the input is less than five), and then decrease until reaching the saturation point (when input is 13). The reason is that with insufficient amount of inputs in the workload, irrelevant checks could

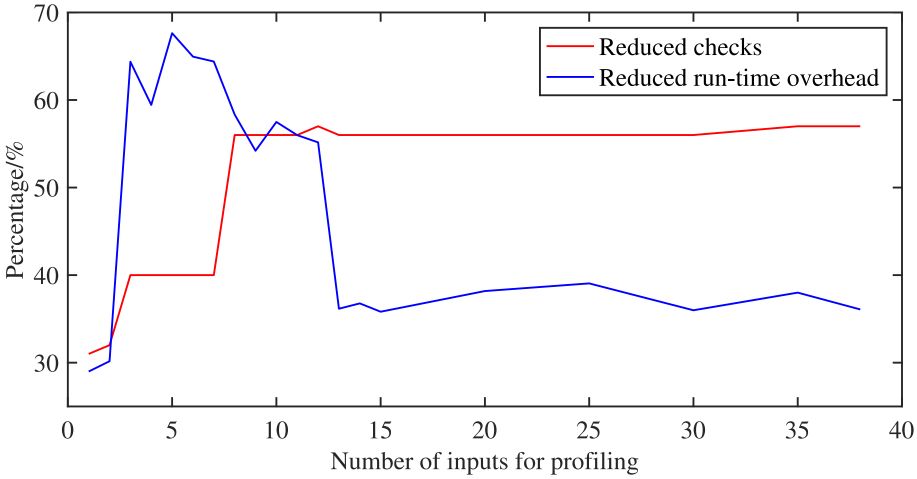

Figure 1: Effects of workload selection evaluation on ASan.

exhibit identical dynamic coverage patterns and be treated as "redundant." In other words, SanRazor could report false positives when the available inputs are insufficient, and aggressively flag too many "redundant" checks (we give discussions on false positives and false negatives of this research in the "Discussion" section of our submitted manuscript). Overall, we interpret that when adopting SanRazor in real-world scenarios, sufficient workload needs to be provided in order to achieve good reduction results and reduce false alarms, but may not need to be too much.

References

[1] Large single compilation-unit C programs. `https://people.csail.mit.edu/smcc/projects/single-file-programs/`, 2006.

[2] Bzip2 testsuite. `https://sourceware.org/git/?p=bzip2-tests.git`, 2019.