Be Sensitive and Collaborative: Analyzing Impact of Coverage Metrics in Greybox Fuzzing

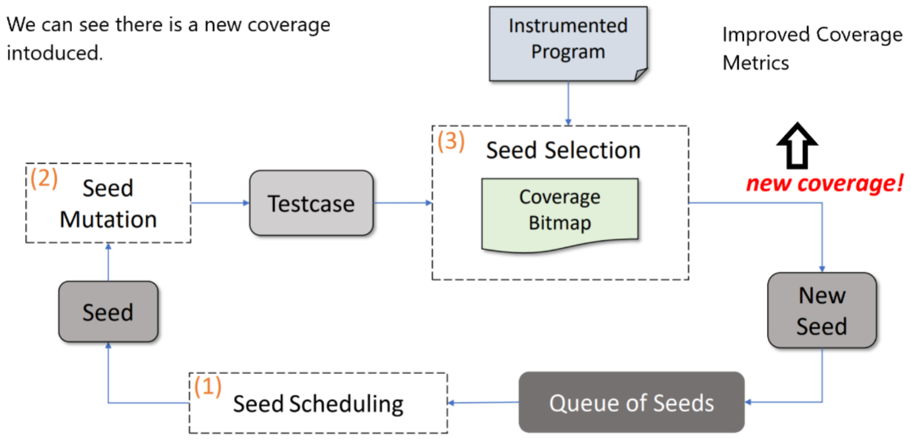
Review

Coverage-based Greybox (CGF) Fuzzing is one of the modernly used techniques in the field of testing in this modern world. Let us first understand a little bit about Greybox Fuzzing what does it mean and how it is helpful in the field of testing. Coverage-based Greybox fuzzing is one of the most successful methods for automated vulnerability detection. Given a seed file (as a sequence of bits), CGF randomly flips, deletes or bits to generate new files. So, when we research as where this technique has been made in use, was able to discover that Greybox fuzzing is a state-of-the-art-program in the field of testing techniques. This technique is mainly adopted by some mainstream companies like Google and Adobe. From some of my research these days many startups try to use Greybox fuzzing these days which they find very helpful.

So what we are trying to analyze from this research paper is how the Coverage metrics can help the Greybox fuzzing. So, when we dive deeper into this research we can see that this contains three main stages they are ***seed scheduling, seed mutation, and seed selection.*** Now from a set of seed inputs there are a lot processing done after which the best and next seed is picked for testing. The seeds are directly proportional to picking and scheduling of the seeds. When we have new seeds that have been selected then there is a more number of new test cases which can perform better and good quality test cases.

What do we mean by Coverage Metrics and how does it impact the entire process is the main thing we need to understand here. Test coverage metrics help you improve the testing process and maximize efficiency. The research tells us that after every test case that is done we get some coverage metrics and improving that metrics will future more help us in selecting the seeds and hence improving for more number of test cases and better test results. We can summarize with a simple image.

Coverage metrics can be categorized into two major categories: code coverage and data coverage. Code coverage metrics evaluate the uniqueness among test cases at the code level, such as line coverage, basic block coverage, branch/edge coverage, and path coverage. Data coverage metrics, on the other hand, try to distinguish test cases from a data accessing perspective, such as memory addresses, access type (read or write), and access sequences.



The image shows us as of where we are introducing the new coverage metrics for the test cases. So how is it done ?.

Experiment:

Experiments are conducted on a private cluster consisting of a pool of virtual machines. As fuzzing is a random process, we followed the recommendations from [20] and performed each evaluation several times for a sufficiently long period. The tests are mainly focused on the CGC dataset. Specifically, each coverage metric is tested with every binary of the CGC dataset in the dataset using two fuzzing instances. Once the experiment is done We comparing different coverage metrics, a central question is "is metric A better than metric B?" To answer this question, we need to look at three main conditions - Unique crashes, Time to crash and Seed count.

Dataset used for this Experiment is We collect binaries from DARPA Cyber Grand Challenge (CGC). There are 131 binaries from CGC Qualifying Event (CQE) and 74 binaries from CGC Final Event (CFE), and thus 205 ones in total. These binaries are carefully crafted by security experts to utilize different kinds of and embed vulnerabilities in various ways to comprehensively evaluate various vulnerability discovery techniques. If we need to know more about the dataset here is a sample: https://github.com/bitsecurerlab/afl-sensitive

We can conclude that from the evaluation results above, we observe that each coverage metric has its unique characteristics in terms of crashes found and crashing times. This observation leads us to wonder whether combining fuzzers with different coverage metrics together would find more crashes and find them faster. So all we can say here is improving the coverage metrics can improve the Seed selections and also for new and better test cases for the same.

Reference:

Wang, Jinghan, et al. "Be sensitive and collaborative: Analyzing impact of coverage metrics in greybox fuzzing." *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*. 2019.