COMS W4995 Secure SW Development

Project Proposal - Coverage Based Fault Injection

Kunal Baweja (kb2896) Siddharth Shah (sas2387) Akshay Nagpal (an2756)

Introduction

We describe a methodology for testing a program for possible security flaws related to its interaction with its environment. Generally, security testing is done using penetration analysis and formal methods. But, most security flaws occur due to unexpected interaction with the environment, and hence it becomes an important criteria for the fault-tolerance properties of a program. Our methodology is based on fault-injection along with code coverage as a parameter for its completeness of test results.

General Approach

Our approach relies to check how the software behaves when there are failures in the interaction with the environment and how gracefully the system handles such cases and look for potential security flaws.

We saw Libfaultinj [2] and have thought of creating a similar tool, extending some of the functionality checks from it and add the ones mentioned below:

- Modify libc functions by making wrappers around them which can simulate unexpected behavior using LD_PRELOAD flag at compile time for code instrumentation:
  1. Malloc
  2. Memory corruption
  3. Network delay
  4. Disk operation delay
  5. Processor affinity (sched_affinity) etc.
- Generate faults that increase test coverage by directing execution towards faulty branches which are rarely executed during normal behavior

Advantage over current approach

The idea of incorporating coverage based testing alongside fault injection testing of programs is based on the intuition that the more lines of code that are executed during testing, the more is the probability of finding crucial security flaws in a program. The current fault injection based programs do not take this into consideration while testing fault tolerance capabilities of a program for security bugs, which we try to demonstrate in this project.

References

1. Vulnerability Testing of Software System Using Fault Injection - https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/98-02.pdf
2. Libfaultinj (Fault injection library) - https://github.com/androm3da/libfaultinj