Papers

1. FUZZIFICATION: Anti-Fuzzing Techniques

Summary:
Fuzzing is a software testing technique which is tests the software automatically by exploring various input space of the software.
But the fuzzy techniques can also be used by attackers to find the bugs in the software and attack it. So to avoid the state of the art fuzzing attacks fuzzification techniques is used so that the attackers can find few bugs and in the meantime the developer team who are running the fuzzers on the unprotected code find and fix all the bugs.

Goals of fuzzification techniques:
1) Hinder existing fuzzing tools and find fewer bugs within a fixed time
2) fuzzified program should still run efficiently in normal use.
3) Protection code should not be easily identified or removed from the protected binary by straightforward analysis techniques.

In this paper evaluation of the fuzzification, technique is tested by maintaining two datasets on is with fuzzification applied on LAVA-M dataset and nine free libraries such as libjpeg, peg, libpng, libtiff, pcre2, readelf, objdump, nm, objcopy, and MuPDF and the other dataset containing the same applications without the fuzzification technique. Also, the author chooses four types of fuzzers AFL, HonggFuzz, VUzzer and Qsym and runs them on two datasets to find that more vulnerabilities are detected when the fuzzification technique is not used. According to the authors this shows that the fuzzification technique is exposed few of vulnerabilities to the attackers.

Fuzzification techniques:
1) SpeedBump:
Provide delays in less used paths using the profiling step (assuming that the attackers will attack the least used path of code)
2) BranchTrap technique used to avoid the creation of unnecessary inputs for all the branches in the program.
3) Anti Hybrid technique disrupts the hybrid fuzzing techniques which attackers are using to find the loopholes in the program.

Pros:

- Fuzzification is quiet flexible which provides various types of configuration options for developers
- Fuzzification is good news for testers as it helps them to catch more issues rather than an attacker or user finding the issues.

Cons:

- Fuzzification causes a slight overhead and code increase and also execution slows down.
- Fuzzification alone cant provides the best security level.
- Fuzzification is a trade-off between providing high security with low performance.

Recent fuzzers list:
AFL
HonggFuzz
QSym- fast concolic execution technique
VUzzer- Utilizes Dynamic taint analysis
T-Fuzz
CSmiths
FuzzIL
Driller: Utilizes symbolic execution
Yarpgen-By intel
CollAFL
REDQUEEN
Dowser

2. Binary Fuzz Testing Method Based on LSTM

Summary:
The author in paper talks about the L-AFL fuzzer(a type of grey box fuzzer) which is coverage based fuzzer that explore all the paths in the target program to reveal the bugs associated with the target program. Also, the author talks about the use of machine learning model called LSTM which learns from the input of the fuzzer running based on the seed provided and the weights of the path executed in the target program by the initial fuzzer.

A fuzzer like AFL is first used with some seed which is nothing but the test data for mutation. Then an Long short Tem memory (LSTM) machine learning model is used to which input is provided in the form of input seed earlier used for mutation and the code coverage path weights of the AFL fuzzer used earlier. The LSTM model further predicts the test data to be used further

for fuzzing on AFL and L-AFL based on the coverage weights the particular test data which should be greater than that of the threshold provided by the user.

The author concludes by saying that the focus of the paper is more towards the mutation of seed and also based on the evaluation the technique call L-AFL achieves higher code coverage than usual AFL.

Pros:

- L-AFL has higher code coverage than AFL on the 4 programs mentioned in the paper.

Cons:

- Not an instant fuzzer. As the LSTM model learns the fuzzing get more and more better so this technique requires more time to get stable and to provide results.