Fuzzer Research

Jack Foley

September 30, 2024

Contents

1 Introduction

This document will outline the research done to start the work on the 4th Software Development Final Year Project (FYP). This project was created by Dr. Chris Meudec and is based on the idea of creating a fuzzer for the C programming language.

2 Fuzzing

Fuzzing is a method of testing software by using broken, random or usual data as an input into the software which is being tested. The idea of fuzzing is that it will find bugs and other issues, such as memory spikes and leaks (temporary denial-of-service), buffer overruns (remote code execution), unhandled exceptions, read access violations (AVs), and thread hangs (permanent denial-of-service).

These are issues that traditional software testing methods, such as unit testing, will not find as easily. There are some different types of fuzzing, such as white-box, grey-box and black-box fuzzing.[3]

2.1 White-box Fuzzing

White-box fuzzing, also known as smart fuzzing, is a technique that is used where the fuzzer is fully aware of the code structure and input variables. White-box fuzzing can lead to discovering bugs more quickly compared to grey-box and black-box fuzzing, but it can also be more computationally expensive as it needs to do an analysis of the codebase before running.

A case study done during development of ISA Server 2006 showed that one defect was found per 17 KLOC (thousand lines of code), a similar black-box fuzzer only found 30% of the defects that the white-box fuzzer found.[3]

2.2 Black-box Fuzzing

Black-box fuzzing is a technique used to test software, analyzing the software by sending random data to the software to discover an application's bugs and vulnerabilities. The black-box fuzzer does not have any information about the inner-workings of the software, it only knows the input and output of the software.

It is a sought-after testing technique as it will work it applications regardless of the programming language or the platform that the software is running on.[1]

2.3 Grey-box Fuzzing

Grey-box fuzzing is a well-known and commonly used fuzzing technique that is used for testing software and finding vulnerabilities. Differing from white-box fuzzing, which can suffer from high computational needs since source code analysis is required, grey-box fuzzing is a very good middle-ground between white-box and black-box fuzzing.[4]

Grey-box fuzzing can also receive coverage feedback from the software, which can then be used to more efficiently traverse the software's codebase to find bugs and vulnerabilities.[2]

References

[1] Aseel Alsaedi, Abeer Alhuzali, and Omaimah Bamasag. "Effective and scalable black-box fuzzing approach for modern web applications". In: *Journal of King Saud University - Computer and Information Sciences* 34.10, Part B (2022), pp. 10068–10078. ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2022.10.006. URL: https://www.sciencedirect.com/science/article/pii/S1319157822003573.

[2] Daniel Blackwell and David Clark. "PrescientFuzz: A more effective exploration approach for grey-box fuzzing". In: (Apr. 2024). arXiv: 2404.18887 [cs.SE].

[3] J. Neystadt. *Automated penetration testing with white-box fuzzing.* Microsoft Learn. Available at: https://learn.microsoft.com/en-us/previous-versions/software-testing/cc162782(v=msdn.10)?redirectedfrom=MSDN (Accessed: 28 September 2024). 2009.

[4] Van-Thuan Pham et al. "Smart Greybox Fuzzing". In: *IEEE Transactions on Software Engineering* 47.9 (2021), pp. 1980–1997. DOI: 10.1109/TSE.2019.2941681.