CERT BFF - Basic Fuzzing Framework

The CERT Basic Fuzzing Framework (BFF) is a software testing tool that finds defects in applications that run on the Linux and Mac OS X platforms. BFF performs mutational fuzzing on software that consumes file input.

Mutational fuzzing is the act of taking well-formed input data and corrupting it in various ways, looking for cases that cause crashes. BFF automatically collects test cases that cause software to crash in unique ways, as well as debugging information associated with the crashes. The goal of BFF is to minimize the effort required for software vendors and security researchers to efficiently discover and analyze security vulnerabilities found via fuzzing.

Traditionally, fuzzing has been very effective at finding security vulnerabilities, but because of its inherently stochastic nature, results can be highly dependent on the initial configuration of the fuzzing system. BFF applies machine learning and evolutionary computing techniques to minimize the amount of manual configuration required to initiate and complete an effective fuzzing campaign. BFF adjusts its configuration parameters based on what it finds (or does not find) over the course of a fuzzing campaign. By doing so it can dramatically increase both the efficacy and efficiency of the campaign. As a result, expert knowledge is not required to configure an effective fuzz campaign, and novices and experts alike can start finding and analyzing vulnerabilities very quickly.

The following are some of the specific features that are available in BFF:

- Minimal initial configuration is required to start a fuzzing campaign.
- Minimal supervision of the fuzzing campaign is required, as BFF can automatically recover from many common problems that can interrupt fuzzing campaigns.
- Uniqueness determination is handled through intelligent backtrace analysis.
- Automated test-case minimization reduces the effort required to analyze results. This is achieved by distilling the test case to the minimal changes to the input data required to induce a specific crash.
- Online machine learning is applied to fuzzing parameter and input file selection to improve the efficacy of the campaign.
- Crash severity/exploitability triage is provided.

At the CERT/CC, we have used the BFF infrastructure to find a number of critical vulnerabilities in products such as Adobe Reader and Flash Player; Foxit Reader; Apple QuickTime, Preview, and Mac OS X; Xpdf; Poppler; FFmpeg; JasPer; Wireshark; VMware VMnc video codec; the Indeo video codec; and many others. See Public Vulnerabilities Discovered Using BFF.

Source Code

Source code for BFF and FOE can be found at at https://github.com/CERTCC/certfuzz.

Issues can be reported at https://github.com/CERTCC/certfuzz/issues.

More information about BFF

- BFF Acknowledgements — Co ntributions to BFF
- BFF Blog Posts
- BFF Installation Notes
- BFF Linux Readme — CERT Basic Fuzzing Framework (BFF) Readme
- BFF Release Notes — CERT Basic Fuzzing Framework (BFF) Significant changes
- Effective use of CERT BFF, or: Brute-force Taint Analysis
- Public Vulnerabilities Discovered Using BFF

Download BFF

Source Code

GitHub

Other Links

- CERT Basic Fuzzing Framework BFF - GitHub
- Samiux's Blog: HOWTO : CERT Basic Fuzzing Framework (BFF) on Ubuntu Desktop 12.04 LTS
- BFF 2.0 ImageMagick Fuzz Run Tutorial - YouTube
- [PDF] Fuzz Testing for Dummies - fuzzing.info
- New CERT Tools Help Developers Find Vulnerabilities
- Guide to setting up BFF using VirtualBox on Ubuntu 10.04