Binary-level Directed Fuzzing for Use-After-Free Vulnerabilities

Manh-Dung Nguyen*[1]

[1]CEA LIST – Université Paris Saclay – France

Résumé

Directed fuzzing focuses on automatically testing specific parts of the code by taking advantage of additional information such as (partial) bug stack trace, patches or risky operations. Key applications include bug reproduction, patch testing and static analysis report verification. Although directed fuzzing has received a lot of attention recently, hard-to-detect vulnerabilities such as Use-Afer-Free (UAF) are still not well addressed, more especially at the binary level. We propose UAFuzz, the first (binary-level) directed greybox fuzzer dedicated to UAF bugs. The technique features a fuzzing engine tailored to UAF specifics, a lightweight code instrumentation and an efficient bug triage step. Experimental evaluation for bug reproduction on real cases demonstrates that UAFuzz significantly outperforms state-of-the-art directed fuzzers in terms of fault detection rate, time to exposure and bug triaging. UAFuzz has also been proven effective in patch testing, leading to the discovery of 20 new bugs in Perl, GPAC and GNU Patch (including a buggy patch) - all of them have been acknowledged and 14 have been fixed. Last but not least, we provide to the community the first fuzzing benchmark dedicated to UAF, built on both real codes and real bugs.

*Intervenant