Taint-Driven Embedded Software Fuzzing

Melisa Savich
New York University
mksavic@nyu.edu

Fuzzing has become a growingly popular method for finding software bugs. Interest in this area of research has been partially reignited by DARPA's Cyber Grand Challenge (CGC) binaries, which focused on creating automatic systems capable of finding flaws and formulating patches [3]. Since then, a multitude of effective and diverse fuzzers have been published like BuzzFuzz, Driller, VUzzer, AFLGo, and Angora. All of these fuzzers have presented powerful techniques in the area of software fuzzing. However, to the best of our knowledge, the area of embedded software fuzzing has yet to be explored.

We attempt to address our problem by building on top of already existing projects. Avatar^2 is an orchestration framework designed to support dynamic analysis of embedded devices [2]. Avatar^2 makes use of PANDA, an open-source Platform for Architecture-Neutral Dynamic Analysis. We make use of one of PANDA's unique features – the ability to record whole system executions of embedded devices [1]. Within PANDA, we write a taint analysis plugin and use it during replays of execution recordings. This plugin helps make informed decisions of which areas of code to fuzz in an embedded system, helping us find meaningful bugs.

References

[1] Brendan Dolan-Gavitt, Joshua Hodosh, Patrick Hulin, Timothy Leek, and Ryan Whelan. *Repeatable reverse engineering with PANDA. In Workshop on Program Protection and Reverse Engineering (PPREW)*, 2015.

[2] Marius Muench, Aurélien Francillon, and Davide Balzarotti. Avatar^2: A Multi-target Orchestration Platform. In *Workshop on Binary Analysis Research (colocated with NDSS Symposium)* (February 2018), BAR 18.

[3] Cyber Grand Challenge (CGC). https://www.darpa.mil/program/cyber-grand-challenge.