FirmFuzz: Automated IoT Firmware Introspection and Analysis

Prashast Srivastava, Hui Peng, Jiahao Li, Hamed Okhravi , Howard Shrobe, Mathias Payer

Writer: Kai Li

1 Summary

This paper proposed an automated device-independent emulation and dynamic analysis framework called FirmFuzz [1] for Linux-based firmware images. It employs a greybox-based generational fuzzing approach coupled with static analysis and system introspection to provide targeted and deterministic bug discovery within a firmware image. Three main features of the FirmFuzz fuzzer are: Context-driven input generation, Deterministic vulnerability detection and Fuzzing side-effects elimination.

- Context-driven input generation: emulation framework, automatically reverts the firmware back to a stable state if the firmware reaches an inconsistent state while being fuzzed.
- Deterministic vulnerability detection: The vulnerability monitors operating both in the guest (i.e., the emulated firmware) and the host allow deterministic vulnerability detection.
- Fuzzing side-effects elimination: FirmFuzz with the help of its emulation framework, automatically reverts the firmware back to a stable state if the firmware reaches an inconsistent state while being fuzzed.

Evaluation results show that FirmFuzz can analyze 32 images (from 27 unique devices) with a network accessible from the host performing the emulation. During testing, FirmFuzz discovered seven previously undisclosed vulnerabilities across six different devices: two IP cameras and four routers and 4 CVE's have been assigned.

2 Strengths and Weaknesses

2.1 Strengths

1. This paper proposed an automated emulation and dynamic analysis framework for finding deep vulnerabilities in embedded firmware, which sheds lights in a new research direction in firmware fuzzing.
2. The authors developed a generational fuzzer for syntactically legal input generation that leverages static analysis to aid fuzzing of the emulated firmware images while monitoring the firmware runtime.
3. Strong evaluation result shows that FirmFuzz can automatically test firmware images scraped from vendor websites and find seven previously unknown vulnerabilities.

2.2 Weakness

1. A major limitation of this work is it cannot completely remove all instances of false negative bugs since FirmFuzz relies on template request generation for fuzzing.
2. Compared with Firmadyne, FirmFuzz is not applicable at large-scale.
3. There could be better (in terms of fairness) testing firmwares to compare FirmFuzz with state-of-the-art works.

References

[1] Prashast Srivastava, Hui Peng, Jiahao Li, Hamed Okhravi, Howard Shrobe, and Mathias Payer. Firmfuzz: Automated iot firmware introspection and analysis. In *Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things*, IoT Samp;P'19, page 15–21, New York, NY, USA, 2019. Association for Computing Machinery.