ReplayFuzzer: IoT attack replay analysis platform based on IoT virtualization

Hye Lim Jung[1] Ki-Woong Park[2*]
[1]SysCore Laboratory,Sejong University, Seoul, Korea
Hyello13@gmail.com
[2]SysCore Laboratory,Sejong University, Seoul, Korea
woongbak@sejong.ac.kr

Abstract

In an IoT device, the sensed value of the sensor measuring the value in real time is input to the firmware, and the operation of the actuator is determined based on the execution of the firmware. Depending on the sensing value of the sensor attached to the IoT device and the stage of the execution process within the firmware, the result of the operation of the IoT device varies significantly. Attackers can exploit these features to artificially cause malfunction in IoT devices. With the development of various sensors and the increasing complexity of sensors, the attack vectors that can be used to exploit the hardware vulnerabilities of sensors are also increasing and will continue to increase in future. In addition, several types of complex sensing devices attached to IoT devices generate asynchronous readings, and multiple sensor nodes cause complex interruptions. Depending on the interrupt handling of the firmware and operation within the routine, the IoT device may inadvertently malfunction, resulting in a potential vulnerability. To overcome these threats, in this study, we propose a ReplayFuzzer that reproduces abnormal behavior by collecting various combinations of sensor data and IoT device-generated interrupt call patterns and emulating them with virtual IoT devices. ReplayFuzzer takes advantage of the fact that the firmware memory layout is fixed after the firmware code is loaded into memory. By repeatedly collecting fixed memory areas of real IoT devices and emulating virtual IoT devices, ReplayFuzzer targets IoT devices equipped with various types of sensors, instead of exploring existing firmware emulation analysis. An effective anomaly analysis process was performed. ReplayFuzzer can be effectively used to derive anomalies in IoT devices, which are becoming more diverse and complex, and to verify the security lines of IoT devices.

Keywords: IoT system, Fuzzing, Virtualization

1 Introduction

With the development of the IoT market, malicious acts that attack IoT have also been developed [1]. An IoT device uses an attached sensor to measure a sensed value in real time and operates as a firmware operation that provides a service to a user based on the value. Firmware execution is based on the value of the sensor attached to the IoT, and an attacker uses this to artificially cause a malfunction[2]. Because sensors attached to the IoT are manufactured by various manufacturers, and there are also different types of sensors, the attack vectors for exploiting hardware vulnerabilities increase proportionally. Such an increase in the attack vector of the sensor module causes the IoT device to malfunction and increases its risk. In addition, various types of complex sensors attached to IoT devices asynchronously compute

The 6th International Symposium on Mobile Internet Security (MobiSec'22), December 15-17, 2022, 2021, Jeju Island, Republic of Korea
Article No. 1, pp. 1-6
*Corresponding author: SysCore Laboratory,Sejong University, Seoul, Korea

readings, and complex interrupts are generated during the process. IoT firmware may also have vulnerabilities when handling interrupts. Depending on the firmware handling method, the interrupts, and the operation within the routine, an IoT device malfunction can be triggered, which can become a potential vulnerability. IoT attacks can lead to serious personal injury and privacy breaches. However, it is difficult to protect against the increasingly advanced vulnerabilities of IoT. The attacked sensor of the IoT device is processed by the IoT algorithm, and the abuse service is provided to the user. This process can be verified in the memory of the IoT devices. With unknown vulnerabilities in IoT sensors, it is difficult for security analysts to detect an IoT attack even if one occurs.

In this study, we propose a ReplayFuzzer platform that analyzes vulnerabilities by collecting sensor information and interrupt call patterns generated by IoT devices and replaying them in a virtual IoT device on a server to reproduce abnormal behavior. Once the IoT firmware is loaded, the memory area remains unchanged. ReplayFuzzer takes advantage of the fact that the firmware memory layout is fixed after the firmware code is loaded into memory. Therefore, ReplayFuzzer repeatedly collects the fixed memory area of the IoT device and performs the same fixed memory dump on the virtual IoT device on the server. It is possible to perform an effective anomaly analysis process for IoT devices equipped with different types of sensors than in studies using existing firmware emulation analysis [3]. The ReplayFuzzer platform proposed in this study can trace the process by reproducing the memory traces left by all attack processes that occur on the IoT device in the virtual IoT device. This can provide insights to security analysts. Moreover, it is expected that ReplayFuzzer can be effectively used to derive abnormal behaviors and verify the security of IoT devices that are constantly changing and becoming more complex.

Other studies [4],[5], [6],[7] have demonstrated the emulation potential of IoT devices, and fuzzers have also been used in studies. The platform proposed in this study focuses on facilitating the analysis of unknown malicious behavior by security analysts. This can be verified by replaying the attack process by collecting sensor values, interrupting the calls of the real IoT device, and injecting them into the virtual IoT device from the server. In addition, the value injected into the virtual IoT device can correspond to the attack scenario configured by the security analyst, not the sensor value of the real IoT device, and the corresponding reaction can be assessed. By performing a memory dump of a virtual IoT device on the host server, a memory dump of a real IoT device and a comparative analysis can be performed.

2 Related Work

The ReplayFuzzer platform proposed in this study can analyze an attack and provide an analysis platform for security analysts by injecting sensor values collected from real IoT devices into virtualization and replaying the execution process. This section describes the platform proposed in this study for such IoT fuzzer studies.

Firm-afl[4] performs fuzzing on POSIX-compliant firmware and solves performance bottlenecks that occur during the execution process through augmented process emulation. An avatar [8] constitutes a hybrid execution environment consisting of real hardware that acts as a software proxy between the emulator and real hardware. This study involves injecting input values into the I/O of the emulator, executing commands, and analyzing the results. The platform proposed in analyzes the situation following a security attack in the IoT environment. To analyze unknown vulnerabilities used in an IoT attack situation, the sensor value of the attack situation is injected. In addition, the platform can verify the calculation process of the IoT firmware by modifying and injecting sensor values. Through these attempts, it is possible to analyze the cause of an attack by analyzing the difference between the emulation environment and endpoint device environment in which the problem occurs. Firmcorn[9] optimizes the virtual environment to solve the shortage of throughput and hardware in fuzzing, and performs a weak code search

algorithm by acquiring a fuzzing entry point. The replayFuzzer platform proposed in this study enables the analysis of the software operation process caused by sensor input through a memory dump.

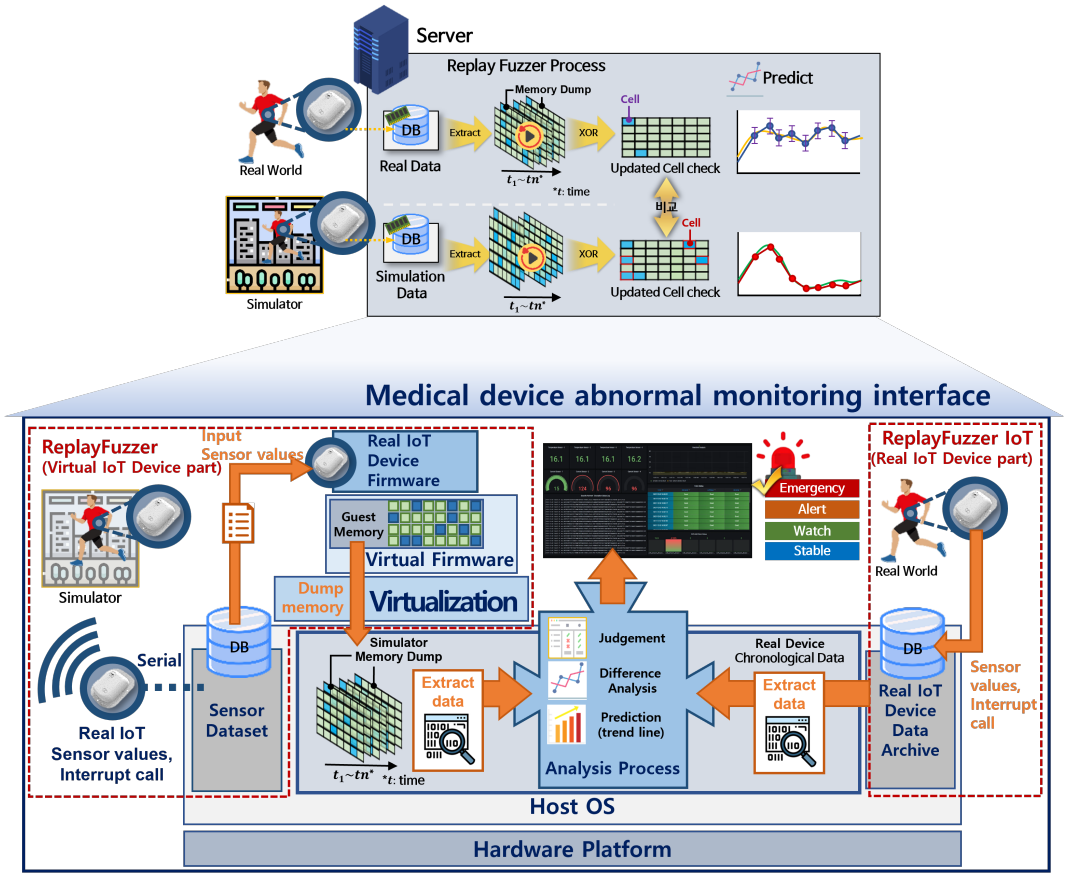

Figure 1: ReplayFuzzer platform design consisting of real IoT devices and virtual IoT devices

3 Implementation of ReplayFuzzer Platform

In Figure 1, the ReplayFuzzer platform consists of real and virtual IoT devices. In a real IoT device, an algorithm must be built into the firmware at the manufacturing stage to transmit the sensor values, interrupt calls, and timestamps generated by the IoT devices to the server. The server collects the sensor values, interrupt calls, and timestamps sent by real IoT devices. The sensor values, interrupt calls, and timestamps collected from a real IoT device are input into the serial communication module of the virtual IoT device. The platform injects sensor value input and interrupts the virtual IoT device using timestamp information. The security analyst can analyze the reaction of the virtual IoT device accordingly. Furthermore, if the security analyst repeatedly dumps the fixed memory area according to the IoT situation and sends it to the server, it can be analyzed later using the fixed memory dump file of the virtual IoT device on the ReplayFuzzer platform. The virtual IoT device was implemented based on virtualization on the server by uploading the same firmware as the real IoT device and configuring it in the same manner as the hardware of the real IoT device. The sensor attached to a real IoT device can be connected to the serial communication module of the virtual IoT device. When the sensor value from the real IoT device is input into the serial communication module of the virtual IoT device, the firmware of

the virtual IoT device performs a service operation based on the sensor value. A module is included in the host OS of the server that performs a memory dump of a virtual IoT device.

The ReplayFuzzer proposed in this study aims to create a simple IoT device with a sensor and a button that detects illuminance and temperature. The data collected by the sensors were transmitted to the server in a 100 ms cycle. The button is the input key that generates an interrupt call. The key sends the sensor data from the IoT device and the resulting value of a simple operation performed on the sensor data to the server. The structure of the data transmitted to the server consists of the sensor value measured by the IoT device, the result of a simple calculation of the sensor value, an interrupt call, and a timestamp. This data is collected on the server. The same firmware as the IoT device is uploaded to the ReplayFuzzer, which is driven in the server to configure the virtual IoT device environment, and the collected sensor values and interrupt call data are input to the virtual firmware of the ReplayFuzzer via the serial communication module. The sensor value and interrupt of the call signal input to the serial communications module can be confirmed by the memory dump operation in ReplayFuzzer. The sensor running in the ReplayFuzzer IoT simulator was also implemented to receive values from the sensor dataset. The simulator's sensor dataset can be run as a dataset and controlled in the simulator by using a value received from an actual sensor or by using the sensor data obtained by analyzing a memory dump collected from an actual device. The memory collected by the ReplayFuzzer simulator is shown in Figure 2, and the injected information is written to memory.

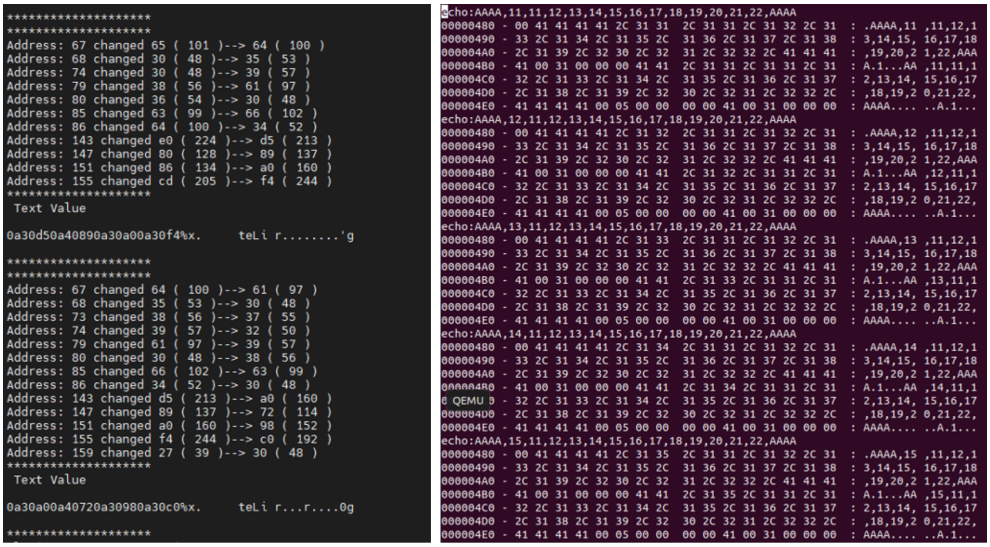

Figure 2: ReplayFuzzer platform design consisting of real IoT devices and virtual IoT devices

This process is performed using a simple IoT device consisting of a sensor and button. The memory dump for a high performance IoT device with complex operations must be customized in detail based on the dump cycle and interrupt call. Real IoT devices have different sensor driving methods and sensor data collection cycles depending on the manufacturer; therefore, advanced analysis requires fine-tuning the calculation process according to the IoT devices. This is done along with experiments on the design for the acceleration of memory dumps and detailed adjustment of sensor data collection cycles, so that it can be practically applied in future studies. Accordingly, based on the IoT device constructed in this study, we propose a ReplayFuzzer platform that can apply a memory dump data replay and perform fuzzing operations that security experts can use for analysis.

4 Conclusion and Future Work

In this study, we presented the ReplayFuzzer platform for security analysts to easily analyze unknown malicious behaviors that may occur in IoT. Unknown malicious behaviors focus on what emanates from the sensors and interrupts attached to IoT devices. The types of sensors attached to IoT devices vary from manufacturer to manufacturer, and are vulnerable to advanced physical attacks. The ReplayFuzzer platform contains a server that collects sensor values, interrupt calls, and timestamps from real IoT devices. The server creates a virtual IoT device with the same hardware and firmware uploaded as the real IoT device. The virtual IoT device receives sensor values and interrupts calls from the real IoT device through a serial communication module. Thus, the reactions of virtual IoT devices can be used by security analysts for analysis. In addition, the ReplayFuzzer platform can replay an IoT device when an abnormal sensor value is input through a virtual IoT device and can analyze unknown vulnerability attacks. In addition, the security analyst can analyze the reaction of the IoT device by adjusting the sensor-value input to the virtual IoT device. It is expected that the ReplayFuzzer can be effectively used to derive abnormal behaviors and verify the security of IoT devices that are constantly changing and becoming more complex.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) , South Korea(Project No. 001657941G0003072, Development of a Multi-Faceted Collection-Analysis-Response Platform for Proactive Response to Ransomware Incidents, 40%), Institute of Information & communications Technology Planning & Evaluation (IITP) (Project No. 2019-0-00426, 10%), Institute of Information & communications Technology Planning & Evaluation (IITP) national defense ICT confluence (national defense information communication network - commercial network(5G), Development of security technology for interworking, No.2022-0-007010001003, 40%) and the ICT R&D Program of MSIT/IITP, South Korea (Project No. 2021-0-01816, A Research on Core Technology of Autonomous Twins for Metaverse, South Korea, 10%)

References

[1] Miao Yu, Jianwei Zhuge, Ming Cao, Zhiwei Shi, and Lin Jiang. A survey of security vulnerability analysis, discovery, detection, and mitigation on iot devices. *Future Internet*, 12(2):27, 2020.

[2] S Pal, M Hitchens, T Rabehaja, and S Mukhopadhyay. Security requirements for the internet of things: A systematic approach. *Sensors*, 20(20):5897, 2020.

[3] C Wright, W A Moeglein, S Bagchi, M Kulkarni, and A Clements. Challenges in firmware re-hosting, emulation, and analysis. *ACM Computing Surveys (CSUR)*, 54(1):1–36, 2021.

[4] Yaowen Z, A Davanian, H Yin, Chengyu S, Hongsong Z, and Limin S. {FIRM-AFL}:{High-Throughput} greybox fuzzing of {IoT} firmware via augmented process emulation. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1099–1114, 2019.

[5] Jiongyi C, Wenrui D, Qingchuan Z, Chaoshun Z, Zhiqiang L, XiaoFeng W, Wing Cheong L, Menghan S, Ronghai Y, and Kehuan Z. Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing. In *NDSS*, 2018.

[6] N Klingensmith and S Banerjee. Hermes: A real time hypervisor for mobile and iot systems. In *Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications*, pages 101–106, 2018.

[7] Bo Yu, Pengfei W, Tai Y, and Yong T. Poster: Fuzzing iot firmware via multi-stage message generation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2525–2527, 2019.

[8] Jonas Z, Luca B, Aurelien F, Davide B, et al. Avatar: A framework to support dynamic security analysis of embedded systems' firmwares. In *NDSS*, volume 14, pages 1–16, 2014.

[9] Zhijie G, Hui S, Fei K, and Xiaobing X. Firmcorn: Vulnerability-oriented fuzzing of iot firmware via optimized virtual execution. *IEEE Access*, 8:29826–29841, 2020.