Blackbox fuzzing approach in Obfuscated Apps for IoT Device

Yeongjae Seo, Dayoung Kim, MinJeong Kang, Seungmin Lee, and Hyo Jin Jo

Soongsil University, Seoul, South Korea
ssyyjj1012@naver.com, dayoung8893@daum.net, codeliq@gmail.com
ares990907@naver.com, hyojin.jo@ssu.ac.kr

Abstract

With the widespread use of various IoT devices in everyday life, research is being proposed for black-box fuzzing as an effective method to detect and respond to vulnerabilities in IoT devices without the need to acquire and analyze their firmware. Previous studies primarily focused on non-obfuscated apps, leaving limitations in detecting vulnerabilities in obfuscated ones. In this work, we propose a novel black-box fuzzing approach that effectively identifies fuzzing Candidates in obfuscated apps and utilizes them.

1 Introduction

As IoT technology grows in importance in modern society, revolutionizing various industries. these devices interact with users and control their functions through companion apps. However, as the use of IoT devices increases, security vulnerabilities in these devices are becoming a significant concern. These security issues threaten users' personal information and impact the devices' stability and reliability.

Existing studies like IoTFuzzer[1] and Diane[2] proposed a black-box fuzzing methodology for IoT devices, but their focus on non-obfuscated apps presents limitations in detecting vulnerabilities in obfuscated ones. The sendMessage function is used to send messages and control the behavior of IoT devices. The Companion App can identify this function by filtering based on the activity for which the network packets are generated. In this poster, we propose an approach to identify sendMessage candidates by filtering the method based on the activity in which network packets occur through the app's UI Activity Tracking in order to perform black box fuzzing for obfuscated apps.

2 Our Approach

We propose a black-box fuzzing approach that aims to enhance IoT device security by effectively identifying the sendMessage function in obfuscated apps. Our approach involves static and dynamic analysis of the application. First, we identify the network communication activity through UI Activity Tracking of the running companion app and filter the methods based on the identified activity. By hooking the filtered method into pysoot, we identify sendMessage candidates for fuzzing and then dynamically connect the candidate method to run the app Once the network traffic is verified, register the last executed sendMessage candidate method. Select the sendMessage candidate that belongs to the cluster with the smallest average elapsed time. Then, to find the fuzzing trigger, we first identify the data transforming function applied to the data being sent. By identifying and controlling the top-level chain functions that affect the sendMessage variable, you can control the data sent to the analyzed IoT device. This top-level chain function is the optimal fuzzing trigger to stimulate the IoT device function. An approach

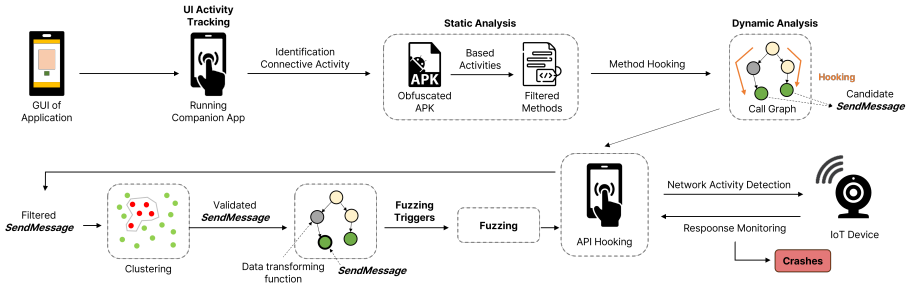


Figure 1: Architecture of the Proposed system

for identifying conflicts in network-based services on IoT devices by fuzzing with corresponding fuzzing triggers. Fig. 1 shows the architecture of the proposed system. This approach will make a significant contribution to the accuracy and efficiency of detecting vulnerabilities in obfuscated apps, its applicability to a wide range of protocols and devices, and the enhancement of IoT security.

3 Conclusion

This study proposes effectively identifying the sendMessage Function in Obfuscated Apps. The sendMessage identified by this approach will allow effective fuzzing with or without obfuscation by the companion app. The proposed approach aims to address the challenges faced by existing research in detecting vulnerabilities in obfuscated applications across diverse protocols and devices, potentially bolstering IoT security and broadening its applicability in various industries.

Acknowledgments

This work was supported by the Technology Innovation Program (P0023522, HRD Program for Industrial Innovation) funded By the Ministry of Trade, Industry Energy(MOTIE, Korea)

References

[1] Qingchuan Zhao Chaoshun Zuo Zhiqiang Lin Xiaofeng Wang W. Lau Menghan Sun Ronghai Yang Kehuan Zhang Jiongyi Chen, Wenrui Diao. Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing. In *Network and Distributed Systems Security (NDSS) Symposium 2018*, pages 1–15. NDSS, 2018.

[2] Nilo Redini, Andrea Continella, Dipanjan Das, Giulio De Pasquale, Noah Spahn, Aravind Machiry, Antonio Bianchi, Christopher Kruegel, and Giovanni Vigna. Diane: Identifying fuzzing triggers in apps to generate under-constrained inputs for iot devices. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 484–500. IEEE, 2021.

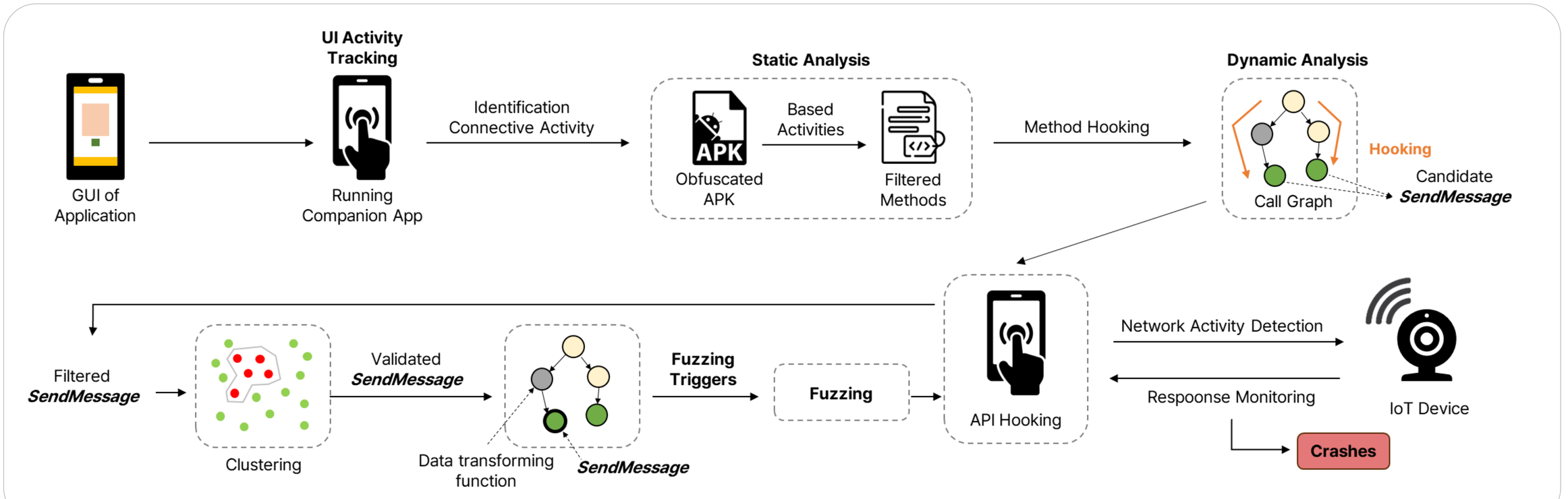
Blackbox Fuzzing Approach in Obfuscated Apps for IoT Devices

Yeongjae Seo, Dayoung Kim, MinJeong Kang, Seungmin Lee, Hyo Jin Jo
Soongsil University, Seoul, South Korea

Introduction

IoT technology's growing importance in modern society has revolutionized various industries. These IoT devices interact with users and are controlled through companion apps. However, as the use of IoT devices increases, so do concerns about security vulnerabilities that threaten personal information and device stability. Existing studies like IoTFuzzer[1] and Diane[2] have proposed black-box fuzzing methods for IoT devices, but they struggle to detect vulnerabilities in obfuscated apps. The 'sendMessage' function is crucial for controlling IoT devices, identifiable by the Companion App through network packet filtering. Our poster introduces an approach to identify 'sendMessage' candidates based on network packet activity, enabling black-box fuzzing for obfuscated apps.

Our Methodology



Methodology Overview

Static and Dynamic Analysis

- Static analysis focuses on the inherent code structure
- dynamic analysis observes runtime behavior

Network Communication Tracking

- Utilize UI Activity Tracking to monitor real-time network communication in the companion app.
- Filter methods based on the activity linked to network packet generation.

Method Identification and Hooking

- Hook the filtered methods using pysoot to identify potential sendMessage candidates.
- These candidates are dynamically connected to the running application for further analysis.

Verification and Selection

- Verify network traffic to confirm the sendMessage function.
- The candidate with the shortest average elapsed time is chosen for efficient fuzzing.

Fuzzing Trigger Identification

- Identify the data transformation function influencing outgoing data and control top-level chain functions the sendMessage variable.

Optimal Fuzzing Trigger

The top-level chain functions, by impacting the sendMessage variable, serve as the best trigger for fuzzing, simulating the device functions.

Conflict Identification

Implement a strategy to identify conflicts in network-based services through dedicated purging triggers.

Contribution

Our approach is set to substantially improve the detection of vulnerabilities in obfuscated apps, with broad applicability across protocols and devices, enhancing overall IoT security.

Conclusion

- This study proposes an effective method for identifying the sendMessage function in obfuscated apps, enhancing security.
- The identified sendMessage function allows for effective fuzzing, regardless of companion app obfuscation, enhancing security testing.
- By addressing challenges in obfuscated app vulnerability detection, this approach has the potential to boost IoT security in diverse industries.

This work was supported by the Technology Innovation Program (P0023522, HRD Program for Industrial Innovation) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)