Poster Abstract: Optimizing IoT Cross-rule Vulnerability Detection through Reinforcement Learning-Based Fuzzing

Tran Ngoc Bao Huynh
nhuynh1@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Ting Xu
ting.xu001@umb.edu
University of Massachusetts Boston
Boston, Massachusetts, USA

Yinxin Wan
Yinxin.Wan@umb.edu
University of Massachusetts Boston
Boston, Massachusetts, USA

Jun Dai
jdai@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Xiaoyan Sun*
xsun7@wpi.edu
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Abstract

Internet of Things (IoT) devices have become increasingly ubiquitous and essential to daily life. These devices are usually controlled based on trigger-action rules, meaning that the devices will take actions according to the rules when trigger conditions are satisfied. As more devices are deployed in smart home systems, the risk of undesirable interactions and cross-rule vulnerabilities increases. In this paper, we propose a reinforcement learning-based fuzzing approach that can automate the modification of environmental variables to generate test cases and increase the likelihood of discovering cross-rule conflicts in smart home systems. Our approach optimizes conflict detection and discovers hidden conditions that lead to vulnerabilities. The preliminary results show that our model can successfully recognize different types of rule conflict.

CCS Concepts

• **Security and privacy → Embedded systems security**.

Keywords

IoT, Fuzzing, Vulnerability Detection, Reinforcement Learning

ACM Reference Format:
Tran Ngoc Bao Huynh, Ting Xu, Yinxin Wan, Jun Dai, and Xiaoyan Sun*. 2025. Poster Abstract: Optimizing IoT Cross-rule Vulnerability Detection through Reinforcement Learning-Based Fuzzing. In *The 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys '25), May 6–9, 2025, Irvine, CA, USA.* ACM, Irvine, CA, USA, 2 pages. https://doi.org/10.1145/3715014.3724024

1 Introduction

The widespread adoption of IoT devices brings significant safety, security, and privacy challenges. As cyber threats and exploitation continue to rise, addressing security vulnerabilities within the IoT ecosystem has become a critical priority. Various smart home systems and IoT devices are controlled with user-trigger rules. When individually deployed, these rules are usually safe and function as intended. However, in an environment with multiple devices that interact with each other, such as in a smart home setting, these rules could cause undesirable or unsafe conflicts [2, 5]. These conflicts could lead to energy inefficiencies, security risks, and even life-threatening situations such as a fire, disaster, hospital setting, etc. For example, *"close the door when it is raining"* and *"open the door if the fire alarm is activated"* are two rules that may cause conflicts. When there is a fire inside, the rule to open the door will be activated. If it is also raining outside, the rule to close the door will also be applied. If the rule to open the door activates before the rule to close, people could get stuck inside and cannot escape.

In this paper, we design and implement a reinforcement learning based fuzzing approach to improve the detection of trigger-action rule conflicts. Instead of manually and randomly feeding different combinations of environmental variables, this approach could autonomously generate test cases and optimize conflict exploration.

2 System Design

The recent survey by Huang [3] categorized IoT conflicts into four main categories: actuation conflict, preference conflict, state impact conflict, and environment conflict. Environment conflict has two subtypes: direct and indirect environmental impact conflict. In this work, we will focus on actuation and environmental conflicts since they usually happen due to changing one or more environmental properties (such as temperature, humidity, weather, etc.). Other types of conflicts arise from human preferences or changes in the system's state. Consider the cases in Figure 1, the "Actuation Conflict" can occur because users could be locked inside when there is fire or smoke, if R2 is activitated before R1. In the case of direct environment conflict, each rule makes sense by itself. However, as the temperature fluctuates around 23°C and 25°C, the heater and the AC may alternate frequently, causing inefficient energy usage. Even worse, if the rules are changed to "*Turn on AC if temperature above 23° C*" and "*Turn on heater if temperature below 25° C*" respectively, both the AC and heater will be turned on when the temperature is around 24°C. Indirect environment conflict is usually more difficult to detect since two rules might seem irrelevant to each other. However, their end actions will affect the same physical channel (i.e., temperature), causing energy waste. For example, if conditions

* Xiaoyan Sun is the corresponding author.

SenSys '25, Irvine, CA, USA

ACM ISBN 979-8-4007-1479-5/25/05
https://doi.org/10.1145/3715014.3724024

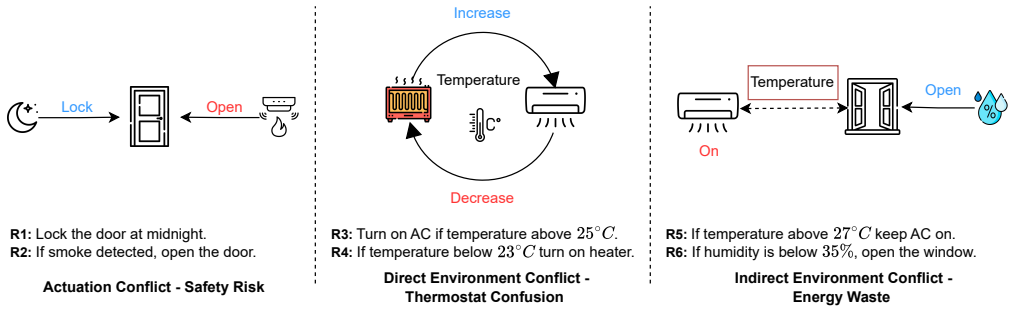

Figure 1: Illustration of different trigger-action rules conflict

are met for both R5 and R6, window will open with the AC turned on. Detecting conflicts in dynamic environments is not straightforward, as a chain of actions is influenced by multiple environmental variables, leading to undesirable interactions. Therefore, it is necessary to test and monitor the system under different combinations of environmental values. Our proposed system consists of three key components, as shown in Figure 2: *Trigger-Action Rule Parser Module, Reinforcement Learning (RL) Fuzzing, and Conflict Detection Module*. The rule parser module will parse user descriptions into trigger-action rules and associated environmental variables.

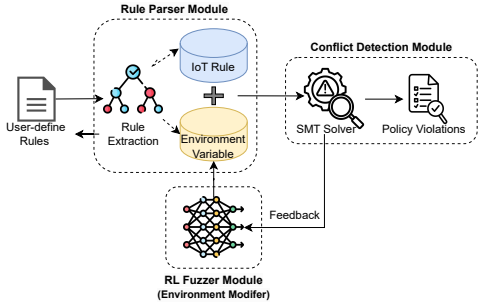

Figure 2: Overview of proposed framework

To ensure proper testing of the fuzzer module, we currently manually parse the descriptions to guarantee the correctness of rule extraction. Future studies will apply NLP (Natural Language Processing) for this task to increase the efficiency, as shown in [4, 6]. The RL Agent employs policy-gradient reinforcement learning to explore the environment by intelligently modifying environmental parameters while monitoring conflicting rule executions. Feedback received from the conflict detection module will help guide the agent to the next action that is most likely to maximize the chances of exposing conflict conditions. The Conflict Detection Module continuously evaluates sensor data and rule executions, identifying and logging inconsistent or contradictory actions based on predefined automation rules. We apply the Z3 Satisfiability Modulo Theories (SMT) solver [1] to analyze the logical relationship between the extracted rules. The SMT Solver is a tool used to determine whether a logical formula is satisfiable (True) or not (False). Previous studies have utilized this tool to identify conflicts among IoT rules [3, 6]. If a rule conflict is detected, the system records the conflicting rules, environmental conditions, and affected devices for further analysis.

3 Preliminary Study

To evaluate the framework, we first provide 12 user trigger-action rules, including three devices (window, AC, and heater) and six environmental variables (room temperature, humidity, smoke, outdoor temperature, rain, and wind speed). The model successfully recognized different types of conflicts shown in Figure 1. For example, given the 21 km/h windspeed and humidity of 29%, the rule "Open the window when humidity below 30%" will conflict with "If the windspeed is over 20km/h, close the window." We observe that the model is better at detecting conflicts with a higher likelihood of occurring. For instance, when considering R5 and R6, if the temperature is maintained at 28°C, the model will capture numerous instances of the same conflict type while mutating the humidity to levels below 35%. This is because we assign the same reward values to each found conflict. To mitigate this, we will update the reward function and assign weight to encourage revealing different conflict types, including conflicts with a low chance of occurring.

4 Conclusion

We proposed a framework for detecting conflicts in trigger action rules by mutating environmental variables. This approach discovers corner cases that lead to interaction conflicts between different IoT devices and suggest possible ways to prevent them.

Acknowledgments

Drs. Xiaoyan Sun and Jun Dai are supported by NSF DGE-2409851. Dr. Jun Dai is also supported by NSF DGE-1934285/2403603.

References

[1] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: an efficient SMT solver. In *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems.*
[2] Jie Hua, Haoxiang Yu, Sangsu Lee, Hamin Md Adal, Colin Milhaupt, Gruia-Catalin Roman, and Christine Julien. 2022. CoPI: Enabling Probabilistic Conflict Prediction in Smart Space Through Context-awareness. In *IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI).*
[3] Bing Huang, Dipankar Chaki, Athman Bouguettaya, and Kwok-Yan Lam. 2023. A Survey on Conflict Detection in IoT-based Smart Homes. *ACM Comp. Surv.* (2023).
[4] Bing Huang, Chen Chen, Kwok-Yan Lam, and Fuqun Huang. 2024. Proactive Detection of Physical Inter-rule Vulnerabilities in IoT Services Using a Deep Learning Approach. arXiv:2406.03836
[5] Muslum Ozgur Ozmen, Xuansong Li, Andrew Chu, Z. Berkay Celik, Bardh Hoxha, and Xiangyu Zhang. 2022. Discovering IoT Physical Channel Vulnerabilities. In *Proc. of ACM CCS.*
[6] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A. Gunter. 2019. Charting the Attack Surface of Trigger-Action IoT Platforms. In *Proc. of ACM CCS.*