Poster: PLUG&CHECK: Finding Bugs in ISO15118 Implementations with EVFUZZ

Ashwin Nambiar*, Z. Berkay Celik*, Ryan Gerdes†, and Antonio Bianchi*
*Purdue University †Virginia Tech

***Abstract*—The ISO15118 standard defines architecture and protocols for V2G (vehicle-to-grid) communication enabling interaction between vehicle manufacturers, utility providers, and charging station operators. This standard was designed to make the charging process easier and secure. However, the security of its implementations has never been evaluated thoroughly. To address this issue, we design a custom system to interact with the car pretending to be a legitimate charger and fuzz the car to find bugs in the implementation of the standard.**

Introduction. The transportation industry adopts Electric Vehicles (EVs) as the major form of transportation. To make this process as smooth and as easy as possible, the industry devised the ISO15118 standard [2]. This standard allows the various manufacturers of the different components of the charging infrastructure to follow a unified standard for authentication, payment verification, and charging. Given the complexity of the standard, the possibility of having bugs in the implementations increases. To address the challenge of effectively fuzzing the ISO15118 stack and identifying bugs, we introduce EVFUZZ. EVFUZZ's charger consists of the Raspberry Pi 3B+, connected to an HSM containing the required certificates to use TLS to communicate with the car using PLUG&CHARGE. Preliminary tests ran the EVFUZZ for 4 generations over 24 hours in 2 modes, observing that most time was spent resetting the car. We employ the concept of unique messages in the car's responses to determine if new paths were discovered.

Implementation. We implemented the fuzzer in Rust as a modification to Nautilus fuzzer [1], as illustrated in Figure 1. To establish communication with the car, EVFUZZ functions as a charging station. To accomplish this, we developed a charging station using ISO15118, Raspberry Pi 3B, and QCA700 Modem. The QCA700 modem was employed to establish Powerline Communication between the charger and the Battery Electric Vehicle (BEV). Additionally, we modified an existing CCS (Combo 2) charger to establish a connection between the Modem and the car by connecting wires. The Modem is controlled through SPI communication using the Raspberry Pi. To ensure we are compatible with PLUG&CHARGE, we obtain a CPO leaf certificate.

Targets. We applied EVFUZZ to three distinct target systems sequentially. We begin with *Java Client (RISE-V2G)* [3], a Java-based client, then with *Python Client (iso15118)* [2], an alternative Python-based client. Finally, we test it on the *Mustang Mach-E*, which serves as the hardware implementation.

Input Generation using Grammar. The fuzzing loop commences by generating inputs based on a specific input grammar

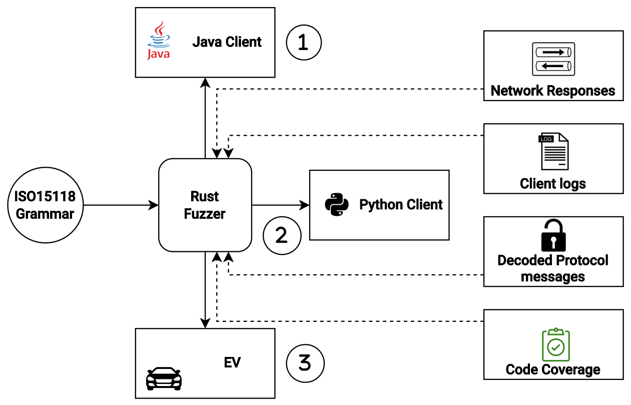

Fig. 1: An overview of the EVFUZZ

(implemented based on [1]), which is derived from the ISO 15118-2 protocol messages.

Fuzzing Loop. To initiate the fuzzing session, the fuzzer generates the initial 10 inputs. In the first round of fuzzing, these inputs are executed on three targets. These three targets offer a range of feedback mechanisms. The fuzzer subsequently acquires feedback from these targets.

Preliminary Results. We ran the EVFUZZ for 4 generations, in 2 modes approximately for 24 hours. During our tests, most of the time was spent on resetting the car. We hypothesize that this is because of an error state triggered by the mutated message. On average, the reset mechanism takes 20 minutes. To determine if the EVFUZZ discovered new paths, we employ the concept of *unique messages* retrieved from the car's responses. We focus on identifying those messages that only appear once in the set of car responses. Our results showed that combining multiple forms of feedback resulted in a greater variety of responses than using only feedback from the car.

ACKNOWLEDGMENT

The authors extend their gratitude to Ford Motor Company for providing research vehicles and technical assistance. Additionally, to Hubject for supplying the leaf certificate for testing, and to the lab members, Chase Smaker and Eric Wilson, who contributed to the hardware interfacing.

REFERENCES

[1] C. Aschermann, T. Frassetto, T. Holz, P. Jauernig, A.-R. Sadeghi, and D. Teuchert, "NAUTILUS: Fishing for Deep Bugs with Grammars," in *Network and Distributed System Security Symposium (NDSS)*, 2019.

[2] "GitHub - SwitchEV/iso15118: Implementation of the ISO 15118 Communication Protocol," https://github.com/SwitchEV/iso15118.

[3] "SwitchEV/RISE-V2G," Switch, Aug. 2023.

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.25006
www.ndss-symposium.org