Hijacking Power: Developing an Exploit for EV Chargers

CS4991 Capstone Report, 2025

Thomas Antal
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
yrz2yy@virginia.edu

ABSTRACT
During my internship with Caesar Creek Software, I was tasked with reverse engineering charger for an electric vehicle and developing an exploit to impact its regular function. To accomplish this, I had to both learn how the device functioned normally, obtain a copy of the firmware, and understand how the different modules communicated between each other. Then I had to decompile and analyze the firmware and analyze memory during runtime to find the sections with accessible vulnerabilities. I ultimately created a payload that, when added, disabled the need for authentication with the built in RFID reader and would begin to charge any attached car. Moving forward, it will be important to look for vulnerabilities in the charger's networking capabilities, as some versions can connect to a Wi-Fi or cellular network. This would eliminate the current need for physical access to the device and make delivering the payload significantly easier.

1. INTRODUCTION

Electric Vehicles have seen rapid growth in recent years, with some considering them the future of personal transit. However, Tesla laid off their whole Supercharger team in April 2024, leading to instability in the EV charger market. As a response, websites such as Amazon have been flooded with at-home chargers that are looking to scoop up a portion of Tesla's market share. While well-known companies such as ChargePoint have several models to choose from, these often cost several hundred if not a thousand dollars. Consumers looking for a more budget-friendly option may instead opt for a charger from a lesser-known brand that costs around two hundred dollars.

As a Cybersecurity Analyst intern, I was given a charger from one of these lesser-known brands and told to uncover everything I could about the way it worked. Conducting a reverse engineering project on the EV charger enabled me to uncover underlying vulnerabilities and see how teams coordinate and develop software together.

2. RELATED WORKS

Over the last 15 years, electric vehicles and plug-in hybrids have gone from nonexistent to a significant portion of market share. In 2014, there were approximately one million electric vehicles in the world. In 2020, there were over ten million and the largest company, Tesla, is valued at over a trillion dollars. Even during the coronavirus pandemic and the brief recession that came with it, sales continued to rise (IEA, 2021).

With the boom in the sales of electric vehicles, many have become frustrated with the current state of charging infrastructure in the United States. The cost of a gallon of gas is plastered on a large board outside of every gas station, and prices must remain competitive to see business. However, this is not the case with charging stations. They do not have the cost per kilowatt hour posted on a large board and

it can be unknown until the customer pulls into the station. This combined with broken equipment has resulted in electric vehicle charging being described as the "Wild West" by some (DeLollis, et. al., 2024). Since consumers are frustrated with the current options, many have been making the switch to at-home chargers. There are currently ten times as many private chargers as public ones, and they saw growth of almost 50% from 2022 to 2023 (IEA, 2024). This growth can see the release of sub-par chargers that cannot properly handle the power requirements. Also, rigorous testing must be done to ensure that these units are safe.

3. PROJECT DESIGN

Beginning a reverse engineering project requires finding out as much as possible about the system one is given. For the EV charger, this meant I had to open it up, remove the motherboard and write down the serial numbers of every chip or peripheral inside. Then, I found the datasheets and manuals for as many of them as possible. Next, I determined which of the chips was the CPU and how the other chips and peripherals were linked together. The motherboard was powered on a 12V rail with a desktop power supply, as running a 240V line would have been incredibly unsafe. The three major steps of the development cycle were obtaining the firmware, gaining access to the charger's memory, and determining the point of vulnerability I wished to exploit.

3.1 Obtaining the Firmware

After reading the manual and datasheet for the CPU, I learned that it had a main flash for its firmware and serial wire debug (SWD) to communicate with it. Tracing the SWD pinout led to a header that a device could be connected to. The board was then put into a "debug mode" by powering it on after connecting a jumper. This enabled SWD and allowed a connection to be formed with a JLink. Using software for the JLink, the firmware and a full memory map was copied from the chip in blocks into binary files. These were then loaded into Ghidra and decompiled into C using the proper ARM instruction set. The datasheets allowed the blocks to be properly labeled and renamed, which made the code significantly easier to understand. This allowed me to more easily find vulnerabilities.

3.2 Gaining Access to Process Memory

While ghidra had a dump of the RAM, it was functionally useless because I could not determine the point in execution it was taken from. After revisiting the manual, I found that the CPU supported on-chip JTAG debugging. However, the pinouts used for this did not go to a header like SWD. Instead, they went to an unpopulated pad near the SWD header. To rectify this, wires were soldered to the pad to allow a connection. This allowed it to connect to an Olimex ARM chip debugger. I then used OpenOCD to run a remote GDB server on the CPU, which allowed me to pause execution, edit registers, and view memory. Physically connecting and disconnecting the Olimex was annoying and had a high likelihood of ripping off the soldered wires. To rectify this, a custom 20 pin header was created to allow for the easy connection and disconnection of the Olimex and Jlink. It also made the motherboard more visually clear, as well as making transporting it safer. I now had full access to the device, making it possible to create an exploit.

3.3 Developing the Exploit

My goal was an exploit that was subtle and affected a peripheral. Logically, this led to the RFID reader as the point of attack. Normally, a user would have to swipe one of the provided RFID cards to unlock the device and enable charging. However, using Ghidra I found that the charger is unlocked when it powers on and locks later in a function call during the bootup sequence. My exploit was a patch in the firmware that wrote out this function call and never locked the device. This would allow anyone to use the device regardless of

authorization. This patch was flashed back into the firmware on the chip.

4. RESULTS

The result was a success. The RFID reader was no longer required. After powering up the board, the charging relay would open up and power would flow through the charger. Repeating this exploit on another charger would be significantly easier and less invasive because it would not be necessary to solder any wires as I did when developing it. All that is required is a JLink and a laptop with the new firmware, making it possible to simply open up the device, put it into debug mode. connect to the SWD header, flash the memory, power cycle the device, and put the cover back on. This full cycle would take less than two minutes in skilled hands, which is important because access time with the device should be minimized for it to be considered a "good" exploit.

This process also yielded other results that are useful if a different exploit is to be developed. Embedded systems often have a checksum that is used to determine if the firmware has been tampered with in any way. This means that any modified firmware could be flashed into the device and be executed without any issues. This also means that other security systems, such as the voltage line that the car and charger use to communicate, could be easily modified or overwritten entirely.

5. CONCLUSION

The result of the project can serve as a method of national security. More and more countries are utilizing electric vehicles, including in military operations. Hijacking an enemy's chargers could cripple their forces and lead to their being useless. Constantly leaving the relay open would cause the batteries to depreciate much faster than normal and result in more maintenance being required as well.

This project gave me firsthand experience with the research and development process required to find software vulnerabilities, develop an exploit, and create a payload to get the exploit into the system. I also saw how teams coordinate and relay new findings between each other. Working with exploits reinforced the need to write vulnerability-free code and gave me the insight to avoid common pitfalls that developers run into.

6. FUTURE WORK

Future work on the project would likely require making the delivery of the payload as easy as possible. This would require making a device to combine the JLink and laptop process described in the results section. Another avenue would be to modify the behavior of the RFID reader itself. This would require additional research into the firmware as I was unable to find where and how the cards are read. If this is insecure then invalid cards could be accepted or exploits including remote code execution could be possible.

7. ACKNOWLEDGMENTS

I would like to thank my supervisor, Nicholas Foster for the guidance throughout the whole development cycle. I would also like to thank the entire Atlanta Caesar Creek software team for training me as I learned new suites of software, and for making the process enjoyable and more than just work.

REFERENCES

IEA. (2021). Trends and developments in electric vehicle markets—Global EV outlook 2021—Analysis. IEA. https://www.iea.org/reports/global-ev-outlook-2021/trends-and-developments-in-electric-vehicle-markets

DeLollis, B., & Justice, G. (2024, June 26). The state of EV charging in America: Harvard research shows chargers 78% reliable and pricing like the "Wild West" | Institute for Business in Global Society. Harvard Business

School. https://www.hbs.edu/bigs/the-state-of-ev-charging-in-america

IEA. (2024). Trends in electric vehicle charging—Global EV outlook 2024--Analysis. IEA. https://www.iea.org/reports/global-ev-outlook-2024/trends-in-electric-vehicle-charging